

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-318783

(43)Date of publication of application : 31.10.2002

(51)Int.Cl.

G06F 13/42

G06F 13/38

(21)Application number : 2001-124248

(71)Applicant : DENSO CORP

(22)Date of filing : 23.04.2001

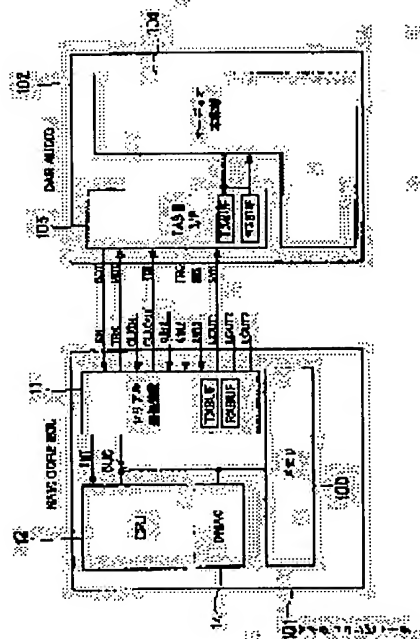
(72)Inventor : KUROYANAGI HITOSHI
RINNAI MASATO

(54) MULTI-PROTOCOL SERIAL COMMUNICATION DEVICE AND MICROCOMPUTER

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a multi-protocol serial communication device which is adaptive to a variety of protocols of serial communication through one chip.

SOLUTION: The multi-protocol serial communication device 11 is so constituted as to generate respective trigger signals of FSTA, FEND, BSTA, and BEND internally by a trigger signal generating means and as those trigger signals are generated, the states of a transmission state machine 38 and a reception state machine 30 are changed to output a necessary output signal to a TAB2 interface part 103 of an ECU101 according to the generation state of a communication control signal.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

【特許請求の範囲】

【請求項 1】 ホストが接続先デバイスとの間でシリアル通信を行う場合に、そのシリアル通信を複数種類のプロトコルに対応して実行可能に構成されるマルチプロトコル型シリアル通信装置であって、ホストによる設定と外部より与えられる通信制御信号の状態とに基づき、各プロトコルに対応したタイミングで、データフレームの開始に対応して出力されるフレームスタート信号、データバイトの開始に対応して出力されるバイトスタート信号、データバイトの終了に対応して出力されるバイトエンド信号、データフレームの終了に対応して出力されるフレームエンド信号、を状態遷移のトリガ信号として発生する送信及び受信トリガ信号発生手段と、前記フレームスタート信号の発生に応じて、フレームスタート待ちステートからバイトスタート待ちステートに遷移し、前記バイトスタート信号の発生に応じて、バイトスタート待ちステートから通信中ステートに遷移することでバイト単位のデータ通信を行わせ、前記バイトエンド信号の発生に応じて、通信中ステートからバイト通信終了ステートに遷移し、バイト通信終了ステートにおいて、前記フレームエンド信号が発生しない場合はバイトスタート待ちステートに遷移し、前記フレームエンド信号が発生した場合はフレーム通信終了ステートに遷移するように構成され、各プロトコル毎に、各ステートに応じて必要な通信制御信号を外部に対して出力させるための送信ステートマシン及び受信ステートマシンとを備えたことを特徴とするマルチプロトコル型シリアル通信装置。

【請求項 2】 前記トリガ信号発生手段は、少なくとも一部のトリガ信号に関する発生条件を選択するためのトリガ条件選択手段を備えており、前記トリガ条件選択手段は、発生条件として、外部より付与される外部トリガ信号を選択可能に構成されていることを特徴とする請求項 1 記載のマルチプロトコル型シリアル通信装置。

【請求項 3】 前記外部トリガ信号は、前記ホストが内部レジスタに書き込みを行うことで発生されるように構成されていることを特徴とする請求項 2 記載のマルチプロトコル型シリアル通信装置。

【請求項 4】 前記トリガ条件選択手段は、前記外部トリガ信号として、前記ホストが内部レジスタに書き込みを行うことで発生させるトリガ信号と、外部より入力ポートに与えられる通信制御信号の遷移状態に基づいて発生されるトリガ信号とを選択可能に構成されていることを特徴とする請求項 3 記載のマルチプロトコル型シリアル通信装置。

【請求項 5】 前記トリガ条件選択手段は、発生条件と

して、通信制御信号の遷移状態をカウントするカウンタにより、そのカウント値が前記ホストによって設定されたカウント値と一致した場合に出力されるカウンタマッチ信号を選択可能に構成されていることを特徴とする請求項 2 乃至 4 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 6】 前記トリガ条件選択手段は、発生条件として、通信データのサイズをカウントするカウンタにより、そのカウント値が前記ホストによって設定されたカウント値と一致した場合に出力されるカウンタマッチ信号を選択可能に構成されていることを特徴とする請求項 2 乃至 5 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 7】 前記トリガ条件選択手段は、受信側のトリガ信号の発生条件として、受信データのパターンが前記ホストより設定された特定のデータパターンに一致した場合にコンパレータより出力されるパターンマッチ信号を選択可能に構成されていることを特徴とする請求項 2 乃至 6 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 8】 送信データ列にビットデータの挿入を行うためのビット挿入手段を備えたことを特徴とする請求項 1 乃至 7 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 9】 前記送信ステートマシンは、フレームスタート待ちステートとバイトスタート待ちステートとの間において、内部レジスタの設定に応じて送信データにフラグを付加するためのフラグデータ挿入ステートに遷移可能に構成されていることを特徴とする請求項 8 記載のマルチプロトコル型シリアル通信装置。

【請求項 10】 前記送信ステートマシンは、バイトスタート待ちステートと通信中ステートとの間において、内部レジスタの設定に応じて送信データにスタートビットを挿入するためのスタートビット挿入ステートに遷移可能に構成されていると共に、通信中ステートとバイト通信終了ステートとの間に、内部レジスタの設定に応じて送信データにストップビットを挿入するためのストップビット挿入ステートに遷移可能に構成されていることを特徴とする請求項 8 または 9 記載のマルチプロトコル型シリアル通信装置。

【請求項 11】 前記送信ステートマシンは、通信中ステートから、内部レジスタの設定に応じて送信データにビットデータを挿入するためのビット挿入ステートに遷移可能に構成されていることを特徴とする請求項 8 乃至 10 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 12】 前記受信ステートマシンは、通信中ステートから、内部レジスタの設定に応じて前記ビット挿入ステートにより受信データに挿入されたビットデータを削除するためのビット削除ステートに遷移可能に構成

10

20

30

40

50

されていることを特徴とする請求項 1 記載のマルチプロトコル型シリアル通信装置。

【請求項 1 3】 前記送信ステートマシンは、通信中ステートとバイト通信終了ステートとの間に、内部レジスタの設定に応じて送信データにパリティビットを挿入するためのパリティビット挿入ステートに遷移可能に構成されていることを特徴とする請求項 8 乃至 1 2 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 1 4】 前記ホストが、外部のメモリと内部の送信バッファ、受信バッファとの間で DMA 転送を行わせるために設けられる、データ転送装置との間で転送制御を行うためのインターフェイス回路を備えていることを特徴とする請求項 1 乃至 1 3 の何れかに記載のマルチプロトコル型シリアル通信装置。

【請求項 1 5】 請求項 1 4 記載のマルチプロトコル型シリアル通信装置と、
ホストとして機能する汎用演算処理装置と、
この汎用演算処理装置がデータの書き込み及び読み出しを行うためのメモリと、
このメモリと前記マルチプロトコル型シリアル通信装置の内部バッファとの間で DMA 転送を行うデータ転送装置とを備えて構成されることを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、シリアル通信を複数種類のプロトコルに対応して実行可能に構成されるマルチプロトコル型シリアル通信装置、及びそのマルチプロトコル型シリアル通信装置を備えて構成されるマイクロコンピュータに関する。

【0002】

【従来の技術】シリアル通信プロトコルには、例えば、RS-232C や IIC (Inter Integrated Circuit), HDLC (High-level Data Link Control procedure), TAB (Telekontrol Audio Bus 2) など、様々な種類がある。ホストたる CPU が、接続先デバイスとの間でシリアル通信を行う場合には、各プロトコルに対応した専用のインターフェイス IC が用意されていたり、或いは UART (Universal Asynchronous Receiver Transmitter) または USART (Universal Synchronous Asynchronous Receiver Transmitter) と称する汎用インターフェイス IC (シリアルポート) を用い、各プロトコルに対応した設定を行なって使用するようにしている。

【0003】

【発明が解決しようとする課題】例えば、図 58 には、ホストのメインプロセッサ (CPU や DSP などの汎用演算装置で構成される) 1 が接続先デバイス 2 との間で何れかのプロトコルに準拠するシリアル通信を実行する場合の構成例を示す。この場合、メインプロセッサ 1 は、接続先デバイス 2 と通信するために汎用ポート (同

期処理や入出力制御を行う) 3 と、上述した例えば USART などのシリアルポート 4 とを直接制御せず、サブプロセッサ 5 を介して通信制御を行うようにしている。

【0004】サブプロセッサ 5 は、メインプロセッサ 1 と同様に CPU, DSP などの汎用演算装置で構成されるが、接続先デバイス 2 について想定される様々なシリアルプロトコルに対してプログラムで柔軟に対応するために配置されている。また、通信処理の実行中には、汎用ポート 3, シリアルポート 4 が頻繁に割り込みを発生させるため、その割り込み処理によってメインプロセッサ 1 の処理効率が低下することを防止する役割をもなしている。しかしながら、斯様な構成では高価なサブプロセッサ 5 を必要とするため、コストアップが避けられない。

【0005】本発明は上記事情に鑑みてなされたものであり、その目的は、サブプロセッサを用いることなくシリアル通信の多様なプロトコルに対応することができ、ホストの通信処理負担を軽減することが可能となるマルチプロトコル型シリアル通信装置、及びそのマルチプロトコル型シリアル通信装置を備えて構成されるマイクロコンピュータを提供することにある。

【0006】

【課題を解決するための手段】請求項 1 記載のマルチプロトコル型シリアル通信装置は、以下の原理に基づいて構成されている。シリアル通信のプロトコルが多様であるとしても、大枠で見ると、それらに共通して行われている原則的な処理方式が存在する。即ち、何らかの形式の信号によって通信の開始と終了を示すと共に、データの送受信は、バイト単位 (通常 8 ビットだが、本発明では定義された複数ビットのまとまりをいう) で行われ、1 回の通信で送受信される複数バイトのデータが 1 つのフレームを構成していると見ることが可能である。

【0007】そして、本発明のマルチプロトコル型シリアル通信装置では、送信または受信トリガ信号発生手段は、上記原則に基づいて、フレーム送受信 (通信) の開始に対応するフレームスタート信号、バイトデータの送受信開始に対応するバイトスタート信号、バイトデータの送受信終了に対応するバイトエンド信号通信、フレーム送受信 (通信) の終了に対応するフレームエンド信号を、ホストによる設定と外部より与えられる通信制御信号の状態とに基づき、各プロトコルに対応したタイミングで状態遷移のトリガ信号として夫々発生させる。

【0008】また、送信及び受信ステートマシンは、フレームスタート待ちステート、バイトスタート待ちステート、通信中ステート、バイト通信終了ステート、フレーム通信終了ステートの各ステート間を遷移するように構成されている。そして、送信及び受信ステートマシンは、フレームスタート信号の発生に応じて、フレームスタート待ちステートからバイトスタート待ちステートに遷移して通信を開始させ、バイトスタート信号の発生に

応じて、バイトスタート待ちステートから通信中ステートに遷移することでバイト単位のデータ通信を行わせる。

【0009】それから、バイトエンド信号の発生に応じて、通信中ステートからバイト通信終了ステートに遷移しバイトデータの送受信に区切りをつけ、バイト通信終了ステートにおいてフレームエンド信号が発生しない場合はバイトスタート待ちステートに遷移してバイトデータの送受信を更に続行する。また、バイト通信終了ステートにおいてフレームエンド信号が発生した場合はフレーム通信終了ステートに遷移して通信を終了させるように動作する。加えて、送信及び受信ステートマシンは、各プロトコル毎に、各ステートに応じて必要な通信制御信号を外部に対して出力させる。

【0010】従って、ホストが例えば内部レジスタ等に必要な設定を行うことで、夫々のプロトコルにおける適切なタイミングでトリガ信号を発生させてステートマシンの状態を遷移させることにより、シリアル通信の多様なプロトコルが実行される場合に容易に対応することが可能となる。

【0011】そして、従来はサブプロセッサ5が行っていた通信処理をシリアル通信装置がハードウェアによって代行することになるので、サブプロセッサ5を使用しない場合でもホストに対して発生させる割り込みを削減することが可能となり、ホストの処理効率が低下することを防止できると共に、ホストを含むシステムを構成する場合のコストを削減することが可能となる。また、各プロトコルに対応してホストに実行させるプログラムを、通信装置に用意されているレジスタの設定に合わせて作成することができるので、プログラミングの手法をある程度標準化させることが可能となり、プログラミング作業をより容易に行なうことができる。

【0012】請求項2記載のマルチプロトコル型シリアル通信装置によれば、トリガ信号発生手段に、少なくとも一部のトリガ信号に関する発生条件を選択するためのトリガ条件選択手段を備え、トリガ条件選択手段を、発生条件として、外部（この場合、ホスト又は接続先デバイス）より付与される外部トリガ信号を選択可能に構成する。即ち、シリアル通信装置の外部より付与される外部トリガ信号を条件としてトリガ信号を発生させることで、例えば、接続先デバイスよりドライブされた通信制御信号の遷移状態をトリガとして、ステートマシンの状態を遷移させることが可能となる。

【0013】請求項3記載のマルチプロトコル型シリアル通信装置によれば、外部トリガ信号が、ホストが内部レジスタに書き込みを行うことで発生されるように構成する。即ち、ホストが制御プログラムに従って動作し、シリアル通信装置の内部レジスタに書き込みを行ったタイミングでトリガ信号を発生させることが可能となる。

【0014】請求項4記載のマルチプロトコル型シリア

ル通信装置によれば、トリガ条件選択手段を、外部トリガ信号として、ホストが内部レジスタに書き込みを行うことで発生させるトリガ信号と、外部より入力ポートに与えられる通信制御信号の遷移状態に基づいて発生されるトリガ信号とを選択可能に構成する。即ち、両者を選択可能とすることで、トリガ信号の発生条件をより柔軟に設定することができ、多様なプロトコル対応を行うことができる。

【0015】請求項5記載のマルチプロトコル型シリアル通信装置によれば、トリガ条件選択手段を、発生条件として、通信制御信号の遷移状態をカウントするカウンタより、そのカウント値がホストによって設定されたカウント値と一致した場合に出力されるカウンタマッチ信号を選択可能に構成する。即ち、斯様なカウンタマッチ信号を選択することで、基準時点から通信制御信号が遷移した回数に基づくタイミングで送信トリガ信号を発生させることができる。

【0016】請求項6記載のマルチプロトコル型シリアル通信装置によれば、トリガ条件選択手段を、発生条件として、通信データのサイズをカウントするカウンタより、そのカウント値がホストによって設定されたカウント値と一致した場合に出力されるカウンタマッチ信号を選択可能に構成する。即ち、斯様なカウンタマッチ信号を選択すれば、基準時点から送信されたデータのサイズ（バイト数）に基づくタイミングで送信トリガ信号を発生させることができる。

【0017】請求項7記載のマルチプロトコル型シリアル通信装置によれば、トリガ条件選択手段は、受信側のトリガ信号の発生条件として、受信データのパターンがホストより設定された特定のデータパターンに一致した場合にコンパレータより出力されるパターンマッチ信号を選択可能に構成する。即ち、斯様なパターンマッチ信号を選択することで、プロトコルで定められている特定のデータを受信した場合に受信トリガ信号を発生させることができる。従って、より多様なプロトコルに対応して適切に受信トリガ信号を発生させることが可能となる。

【0018】請求項8記載のマルチプロトコル型シリアル通信装置によれば、送信データ列にビットデータの挿入を行うためのビット挿入手段を備えるので、例えば、データを送信する際にスタートフラグ、スタートビットやストップビット、パリティビットなどが必要とされるプロトコルにも対応することが可能となる。或いは、送信データ列を適当なビットによって区分することが必要とされる場合にも、適当なデータビットを挿入して対応することができる。

【0019】請求項9記載のマルチプロトコル型シリアル通信装置によれば、送信ステートマシンを、フレームスタート待ちステートとバイトスタート待ちステートとの間において、内部レジスタの設定に応じてフラグデー

タ挿入ステートに遷移可能に構成する。従って、例えば HDLC のように、送信の開始時と終了時にフラグシーケンスを送信する必要があるプロトコルにも対応することが可能となる。

【0020】請求項10記載のマルチプロトコル型シリアル通信装置によれば、送信ステートマシンを、バイトスタート待ちステートと通信中ステートとの間において、内部レジスタの設定に応じてスタートビット挿入ステートに遷移可能に構成し、また、通信中ステートとバイト通信終了ステートとの間に、内部レジスタの設定に応じてストップビット挿入ステートに遷移可能に構成する。従って、例えば RS-232C のように、送信データの最初と最後にスタートビット、ストップビットが場合によって必要となるプロトコルにも対応することができる。

【0021】請求項11記載のマルチプロトコル型シリアル通信装置によれば、送信ステートマシンを、通信中ステートから、内部レジスタの設定に応じて送信データにビットデータを挿入するためのビット挿入ステートに遷移可能に構成する。従って、例えば HDLC のように、ビットスタッフィングを行う必要があるプロトコルにも対応することができる。

【0022】請求項12記載のマルチプロトコル型シリアル通信装置によれば、受信ステートマシンを、通信中ステートから、内部レジスタの設定に応じてビット挿入ステートにより受信データに挿入されたビットデータを削除するためのビット削除ステートに遷移可能に構成する。従って、例えば HDLC のように、ビットスタッフィングが行われているデータから本来不要なビットを削除することができる。

【0023】請求項13記載のマルチプロトコル型シリアル通信装置によれば、送信ステートマシンを、通信中ステートとバイト通信終了ステートとの間に、内部レジスタの設定に応じて送信データにパリティビットを挿入するためのパリティビット挿入ステートに遷移可能に構成する。従って、例えば RS-232C のように、エラー検出用のパリティビットを付加する必要があるプロトコルにも対応することができる。

【0024】請求項14記載のマルチプロトコル型シリアル通信装置によれば、ホストが外部のメモリと内部の送信バッファ、受信バッファとの間で DMA 転送を行わせるために設けられる、データ転送装置との間で転送制御を行うためのインターフェイス回路を備える。

【0025】即ち、通信装置がインターフェイス回路を介してデータ転送装置との間で転送制御を行い、ホストと内部バッファとの間で DMA 転送を行わせるようにすれば、通信装置と接続先デバイスとの間でデータの送受信が行われている間は、通信装置はホストに対して1バイト毎に割り込みを発生させる必要がなくなり、通信が終了した時点で割り込みを発生させれば良い。従って、ホス

ト側の割り込み処理負担を大きく軽減することができる。

【0026】請求項15記載のマイクロコンピュータによれば、例えば、送信の場合は、ホストとして機能する汎用演算処理装置が接続先デバイスに送信するデータをメモリに書き込み、データ転送装置のレジスタに必要な設定を行うと、請求項14記載のマルチプロトコル型シリアル通信装置がデータ転送装置の転送制御を行って、メモリに書き込まれた送信データを内部の送信バッファに DMA 転送を実行させる。また、受信の場合は、接続先デバイスから受信したデータがマルチプロトコル型シリアル通信装置内部の受信バッファに書き込まれると、データ転送装置に受信バッファからメモリに対して DMA 転送を実行させる。

【0027】従って、マルチプロトコル型シリアル通信装置は、汎用演算処理装置に対してバイトデータの送信、受信毎に割り込みを発生させる必要がなく、送信、受信処理の終了時に割り込みを発生させれば良いので、汎用演算処理装置の割り込み処理負担を大きく軽減することができる。

【0028】

【発明の実施の形態】（第1実施例）以下、本発明の第1実施例について図1乃至図22を参照して説明する。図1は、マルチプロトコル型シリアル通信装置の全体構成を概略的に示す機能ブロック図である。マルチプロトコル型シリアル通信装置（以下、単にシリアル通信装置と称す）11は、ホストたる CPU（汎用演算処理装置）12が図示しない接続先デバイスとの間でシリアル通信を行う場合のインターフェイス機能をなすように構成された LSI である。

【0029】シリアル通信装置11は、CPU12が内部のレジスタやバッファなどに書き込み及び読み出しを行うためのホストインターフェイス（I/F）部13を備えている。また、ホスト I/F 部13は、DMA コントローラ（DMAC、データ転送装置）14との間のインターフェイス機能をも含むものである。DMAC14は、図示しない外部メモリとシリアル通信装置11のバッファとの間でデータ転送を行うために設けられている。

【0030】シリアル通信装置11の受信側は、ビット検出部20、クロック生成部21、シフトレジスタ（SHIFTER）22、受信バッファ（BUF）23、一致検出部24、エラー検出及び割り込み制御部25、汎用入力部26並びに受信制御部28などを備えている。

【0031】ビット検出部20は、シリアル受信データ（RX）について伝送符号形式（例えば、NRZ、NRZI、FM0/1、マンチェスタなど）に応じたビット検出を行うように構成されている。クロック生成部21は、復調されたビット列及び外部より与えられるシステムクロック信号（SYSCK）や同期クロック入力信号（CLKI

N) に基づいて同期クロック出力信号(CLKOUT)を生成するものである。

【0032】シフトレジスタ22は、シリアル受信データを格納してパラレル変換するものであり、受信バッファ(内部バッファ)23は、パラレル変換された受信データが格納されるようになっている。エラー検出及び割り込み制御部25は、シリアル受信データに基づいて受信データパターンの一致検出を行ったり、割り込み信号(INT)を発生させる機能を有している。

【0033】汎用入力部26は、接続先デバイスより与えられる通信制御信号が、プロトコルに応じて3つの入力端子UIN1~3に適宜割り当てられるようになっている。また、受信制御部28は、受信コントロール部29、受信ステートマシン30及び受信カウンタ31を備えている。受信コントロール部29は、CPU12が受信側の処理について各種設定を行うためのレジスタ群や後述する受信トリガ信号発生部、パターンマッチ回路などで構成され、受信ステートマシン30は、異なるプロトコルに対する受信処理を共通化してなるステートを受信トリガ信号に応じて遷移させる論理回路である。

【0034】受信カウンタ31は、詳細は後述するが、通信制御信号の遷移状態や受信データサイズなどをカウンタするためのカウンタであり、それらのカウンタ値がレジスタに設定されたカウンタ値と一致すると、カウンタマッチ信号を出力するようになっている。

【0035】一方、シリアル通信装置11の送信側は、送信バッファ(内部バッファ)32、シフトレジスタ33、ビット符号化部34、送信制御部35及び汎用出力部36などを備えている。送信バッファ32は、CPU12またはDMAC14より書込まれる送信データが格納されるようになっており、シフトレジスタ33は、送信バッファ32に格納されたデータをシリアル変換してビット符号化部34に出力するものである。ビット符号化部34は、シリアル送信データ(TX)について前述した伝送符号形式に応じたビット符号化を行うように構成されている。

【0036】送信制御部35は、送信コントロール部37、送信ステートマシン38及び送信カウンタ39を備えている。送信コントロール部37は、CPU12が送信側の処理について各種設定を行うためのレジスタ群や後述する送信トリガ信号発生部などで構成され、送信ステートマシン38は、異なるプロトコルに対する送信処理を共通化してなるステートを送信トリガ信号に応じて遷移させる論理回路である。また、ホストたるCPU12がレジスタに設定を行なうことにより、汎用出力UOUT1~3やクロック生成部21より出力されるCLKOUTの信号レベルをソフトウェア的に制御することも可能となっている。

【0037】送信カウンタ39は、詳細は後述するが、通信制御信号の遷移状態や送信データサイズなどをカウ

ンタするためのカウンタであり、それらのカウンタ値がCPU12によってレジスタに設定されたカウンタ値と一致すると、カウンタマッチ信号を出力するようになっている。汎用出力部36は、接続先デバイスより与えられる通信制御信号が、プロトコルに応じて3つの出力端子UOUT1~3に適宜割り当てられるようになっている。

【0038】図2は、シリアル通信装置11で取り扱われるデータ列及びトリガ信号の定義を示すチャートである。即ち、受信または送信データ(RX, TX)は、Nビット(例えば、N=8)のまとまりをバイトと称し、一回の通信で送受信されるバイトデータのまとまりをフレームと称する(図2(a)参照)。データの通信は、各ビットがクロック信号CLK(図2(b)参照)に同期するようにして行われる。

【0039】そして、通信の開始時に出力される信号FSTAをフレームスタート信号(図2(e)参照)、バイトデータの通信の開始時に出力される信号BSTAをバイトスタート信号(図2(c)参照)、バイトデータの通信の終了時に出力される信号BENDをバイトエンド信号(図2(d)参照)、通信の終了時に出力される信号FENDをフレームエンド信号(図2(f)参照)と称する。

【0040】また、図3は、受信時における各種クロック信号の定義を示すタイミングチャートである(バイト数N=3の場合)。伝送クロックTCKは、データの伝送制御に使用されるクロックであり(図3(a)参照)、受信データRXは、ハイレベルから1ビット期間だけロウレベルに遷移することでデータの送信開始(スタートビット)を示す(図3(b)参照、但し、プロトコルに応じて必要な場合)。そして、3ビットのデータD0~D2を受信した後、1ビットのハイレベル期間でデータの送信終了(ストップビット、必要な場合)を示す。

【0041】FF_Qは、RXのTCK同期信号であり(図3(c)参照)、前述したバイトスタート信号BSTAは、FF_Qのスタートビットと同時に(スタートビットを送信する場合)出力される(図3(d)参照)。そして、バイトエンド信号BENDは、FF_Qのストップビットと同時に(ストップビットを送信する場合)出力される(図3(e)参照)。受信中信号RENGは、その間のデータD0~D2が受信されている間にハイレベルとなって受信中であることを示す信号である(図3(f)参照)。そして、データシフトクロックSCKは、伝送クロックTCKがデータD0~D2の受信中にだけ出力されるクロックであり、受信データのシフトに利用される(図3(e)参照)。

【0042】次に、受信側の構成を中心として説明する。図4は、受信バッファ23を中心とする詳細な構成を示すものである。ビット検出部20で検出された受信

10

20

30

40

50

データ列は、シフトクロック SCK により 16 ビットのシフトレジスタ 22 に格納され、2 つの 32 ビット受信バッファ 23 (0), 23 (1) にパラレルデータとして格納される。ライトポインタによって下位側の受信バッファ 23 (0) から先に格納され、32 ビットモードの場合は 32 ビットが格納された時点で、8 ビットモードの場合は 8 ビットが格納された時点でフルとなり、受信バッファ 23 (1) 側に移る。受信バッファ 23

(0), 23 (1) に格納されたデータは、マルチプレクサ 40 を介して CPU 12 側のデータバス 41 に出力される。

【0043】ビット検出部 20 は、受信データ RX のビット列より生成したクロック BCK をクロック生成部 21 に出力し、クロック生成部 21 は、クロック BCK の出力の有無に応じて CPU 12 側より供給される外部クロック SYSck, 接続先デバイス側より供給される外部クロック CLKIN の何れか一方を伝送クロック TCK として AND ゲート 42 に出力する。AND ゲート 42 には、その他、後述する受信許可信号 RE 及び受信中信号 REN

NG が入力されており、それらの論理積結果としてシフトクロック SCK を出力するようになっている。

【0044】図 5 (a) は、受信側のパターンマッチ回路 43 の構成を示すものである。パターンマッチ回路 43 は、マルチプレクサ 44, シフトレジスタ 46, コンパレータ 47, データ設定レジスタ 48, サンプル区間設定レジスタ 49 及び論理回路 50 などで構成される。8 ビットのシフトレジスタ 46 は、ビット検出部 20 より出力される受信データ列を、マルチプレクサ 44 を介して与えられるクロック TCK, SCK の何れかでシフトするようになっている。

【0045】コンパレータ 47 は、シフトレジスタ 46 に格納された受信データと、データ設定レジスタ 48 にセットされたパターンデータとを比較して一致信号 (PAT_MATCH) を出力する。図 5 (b) に示す論理回路 50 は、フレームスタート信号 FSTA をトリガとして、サンプル区間設定レジスタ 49 に設定されたサンプル開始位置のデータまでカウントを行うとイネーブル信号 SE の出力を開始し、その時点から同レジスタ 49 に設定されたサンプル終了位置のデータまでカウントを行うとイネーブル信号 SE の出力を停止するようになっている。このイネーブル信号 SE をコンパレータ 47 に与えることで、連続するデータの特定部分のみを指定してデータパターンの比較を行わせるようにしている。パターンマッチ回路 43 は、例えばシリアルデータ RX として送信されるアドレス値の比較などに用いられる。

【0046】図 6 は、カウンタマッチ検出回路 51 の構成を示すものである。カウンタマッチ検出回路 51 は、カウンタ 52, カウント値設定レジスタ 53, マルチプレクサ 54 及び 55 などを中心として構成されている。カウント値設定レジスタ 53 に設定されたデータはカウ

ンタ 52 にロードされ、マルチプレクサ 54 によって選択される信号によってダウンカウントが行われる。そして、カウント値が “0” になった時点でカウンタマッチ信号 CNT_MATCH を出力するようになっている。

【0047】マルチプレクサ 54 では、トリガ信号 FSTA, FEND, BSTA, BEND に加えて、伝送クロック TCK 及び後述する同期信号 ACK の内から何れか 1 つが選択される。また、カウンタ 52 のクリアは、OR ゲート 56 を介したカウンタマッチ信号 CNT_MATCH の出力によるか、或いは、マルチプレクサ 55 で選択される上記トリガ信号 FSTA ~ BEND, 及び受信データ RX またはその反転レベルによって行われる。

【0048】図 7 は、受信制御部 28 における受信ステートマシン 30 の状態遷移を示すものである。受信ステートマシン 30 には、概括して以下の 6 つのステート (状態) が規定されている。

- ① フレームスタート待ち
- ② バイトスタート待ち
- ③ 受信中
- ④ バイト受信終了処理
- ⑤ フレーム受信終了処理
- ⑥ エラー

【0049】そして、リセット状態で①のステートとなり、そこから、各トリガ信号の発生によって以下のように遷移する。

```

FSTA: ①→②
BSTA: ②→③
BEND: ③→④
FEND: ④→⑤
not FEND: ④→②
ERR: ④→⑥→①, ⑤→⑥→①
not ERR: ⑤→①

```

尚、トリガ信号 ERR は、図示しないエラー検出回路が CRC や FCC に基づいて受信データにエラーを検出した場合に出力される。また、送信ステートマシン 38 についても、図 7 のレベルでは、⑥エラーを除いて受信ステートマシン 30 と同様に構成される (但し、“受信” → “送信”)。また、ステート⑤において “not ERR” である場合は通信処理を一旦終了しても良い。

【0050】次に、図 8 は、受信コントロール部 29 における受信側のフレームスタート信号 RX_FSTA の出力回路 (トリガ信号発生手段) 57 の構成を示すものである。出力回路 57 は、マルチプレクサ (トリガ条件選択手段) 58 及び AND ゲート 59 で構成されており、マルチプレクサ 58 は、以下の 4 つの信号①~④の内 1 つをトリガ信号として、レジスタ RX_FSTA_SEL の設定に応じて選択的に AND ゲート 59 に出力するようになっている。

① INP1 : 汎用入力 UIN1 / ホストによる S/W

設定

②PATM1 : パターンマッチ 1

③RX_MCNTM : 受信マルチカウンタマッチ

④RE : 受信許可信号

そして、ANDゲート59は、マルチプレクサ58が出力する信号と、受信許可信号REとの論理積をとってフレームスタート信号RX_FSTAを出力する。即ち、マルチプレクサ58において④REが選択された場合には、受信許可信号REの発生のみがトリガ信号の発生条件となる。

【0051】また、①INP1は、別途のレジスタ設定によって汎用入力UIN1に与えられる外部信号と、CPU12が指定されたレジスタに書き込みを行って発生させるトリガ信号(S/W(ソフトウェア)設定)の何れかを選択することが可能となっている。

【0052】図9は、受信側のフレームエンド信号RX_FENDの出力回路(トリガ信号発生手段)60の構成を示すものである。出力回路60は、マルチプレクサ(トリガ条件選択手段)で構成されており、以下の4つの信号①~④の内1つをトリガ信号として、レジスタRX_FEND_SELの設定に応じて選択的に出力するようになっている。

①RX_DCNTM : 受信データサイズカウンタマッチ

②RX_MCNTM : 受信マルチカウンタマッチ

③PATM1 : パターンマッチ 1

④INP1 : 汎用入力UIN1/ホストによるS/W設定

【0053】図10は、受信側のバイトスタート信号RX_BSTAの出力回路(トリガ信号発生手段、トリガ条件選択手段)61の構成を示すものである。出力回路61は、マルチプレクサで構成され、出力回路57のマルチプレクサ58に与えられている3つの信号①~③に加えて、

④INP3 : 汎用入力UIN3/ホストによるS/W設定

が与えられており、レジスタRX_BSTA_SELの設定に応じて選択的に出力するようになっている。また、受信側のバイトエンド信号RX_BENDは、バイトサイズカウンタ(受信トリガ信号発生手段)のマッチ信号がそのまま出力されるように固定されている。

【0054】図11は、送信バッファ32を中心とする詳細な構成を示すものである。CPU12側のデータバス41より転送された送信データは、32ビットの送信バッファ32に格納され、16ビットのシフトレジスタ33に読み出される。32ビットモードの場合はシフトレジスタ33により2回または4回で読み出され(シフトレジスタ33の使用ビット数で決まる)、8ビットモードの場合は1回で読み出される。送信バッファ32のエンプティ信号TBEMPTは、リセット入力時、送信開始時及び送信バッファ32が空になった場合に出力され、D

MAC14に対しては転送要求信号DREQとして出力される。

【0055】シフトレジスタ33によりシリアルに送出される送信データ列は、ビット挿入部(ビット挿入手段)62においてプロトコルに応じてスタートビット、ストップビット、パリティビットなどの挿入が行われてからビット符号化部34に入力される。

【0056】図12は、送信コントロール部37における送信側のフレームスタート信号TX_FSTAの出力回路(トリガ信号発生手段)63の構成を示すものである。出力回路63は、マルチプレクサ(トリガ条件選択手段)64及びANDゲート65で構成されており、マルチプレクサ64は、以下の4つの信号①~④の内1つをトリガ信号として、レジスタTX_FSTA_SELの設定に応じて選択的にANDゲート65に出力するようになっている。

①TE : 送信許可信号

②INP2 : 汎用入力UIN2/ホストによるS/W設定

③INP2+notTBEMPT : ②とバッファエンプティTBEMPTの否定との論理和

④TX_MCNTM : 送信マルチカウンタマッチ

そして、ANDゲート65は、マルチプレクサ64が出力する信号と、送信許可信号TEとの論理積をとってフレームスタート信号TX_FSTAを出力する。即ち、マルチプレクサ64において①TEが選択された場合には、送信許可信号TEの発生のみがトリガ信号の発生条件となる。

【0057】また、②INP2は、別途のレジスタ設定によって汎用入力UIN2に与えられる外部信号と、CPU12が指定されたレジスタに書き込みを行って発生させるトリガ信号の何れかを選択することが可能となっている。

【0058】図13は、送信側のフレームエンド信号TX_FENDの出力回路(トリガ信号発生手段、トリガ条件選択手段)66の構成を示すものである。出力回路66は、マルチプレクサで構成されており、以下の4つの信号①~④の内1つをトリガ信号として、レジスタTX_FEND_SELの設定に応じて選択的に出力するようになっている。

①TX_DCNTM : 送信データサイズカウンタマッチ

②TX_MCNTM : 送信マルチカウンタマッチ

③INP2 : 汎用入力UIN2/ホストによるS/W設定

④INP3 : 汎用入力UIN3/ホストによるS/W設定

【0059】図14は、送信側のバイトスタート信号TX_BSTAの出力回路(トリガ信号発生手段)67の構成を示すものである。出力回路67は、マルチプレクサ(トリガ条件選択手段)68、ANDゲート69及びマルチ

プレクサ 70 で構成されている。マルチプレクサ 68 には、以下の 3 つの信号①～③が与えられており、レジスタ TX_BSTA_SEL の設定に応じて選択的に出力するようになっている。

① INP2 : 汎用入力 UIN2 / ホストによる S/W 設定

② INP3 : 汎用入力 UIN3 / ホストによる S/W 設定

③ TX_MCNTM : 送信マルチカウンタマッチ

そして、マルチプレクサ 68 の出力信号は、AND ゲート 69 を介して、また、直接マルチプレクサ 70 の入力端子に与えられている。AND ゲート 69 の他方の入力端子には送信パツファのエンプティ信号 TBEMPTY の反転が与えられており、マルチプレクサ 70 はレジスタ CLK_MASTER の設定に応じてマルチプレクサ 68、AND ゲート 69 の出力信号を選択的に出力する。また、送信側のバイトエンド信号 TX_BEND は、受信側と同様に、バイトサイズカウンタ（トリガ信号発生手段）のマッチ信号がそのまま出力されるように固定されている。

【0060】図 15 は、シリアル通信装置 11 に汎用的に入力される信号を内部制御に使用するための構成を概略的に示すものである。前述したように、汎用入力信号選択レジスタ INP_SEL の設定に応じて、マルチプレクサ 71 において汎用入力 UIN1～3 と、CPU 12 によるレジスタ INP_SWCTL に対する書込みとの何れか一方を選択して、送受信制御や割り込み要因などに使用することができる。

【0061】また、論理回路 72 は、マルチプレクサ 71 からの出力信号を、エッジで検出（立上がり、立下がり、両エッジ）するか、またはレベル（High, Low）で検出するかを設定するものであり、設定された検出タイプに応じてトリガ信号を発生するようになっている。そのトリガ信号は、受信制御部 28 や送信制御部 35 に出力されたり、CPU 12 に対する割り込み要因として用いられる。

【0062】図 16 は、シリアル通信装置 11 内部で汎用化したトリガ信号等に基づいて、シリアル通信装置 11 より外部に出力される信号を各プロトコルに応じて生成するための構成を概略的に示すものである。4 つのトリガ信号 FSTA, BSTA, FEND, BEND は、マルチプレクサ 73 S, 73 R に与えられており、マルチプレクサ 73 S, 73 R は、夫々出力開始、出力終了のトリガ信号として何れか 1 つを選択的に出力するようになっている。マルチプレクサ 73 S, 73 R の出力信号はマルチプレクサ 74 S, 74 R に直接入力されると共に、インバータゲート 75 S, 75 R を介して入力されている。

【0063】マルチプレクサ 74 S, 74 R の出力信号は、マルチプレクサ 76 S, 76 R の一方の入力端子に与えられており、マルチプレクサ 76 S, 76 R の他方

の入力端子には、カウンタ 77 S, 77 R のカウンタマッチ出力が与えられている。カウンタ 77 S, 77 R には、カウンタ値設定レジスタ 78 S, 78 R に設定されたデータがロードされるようになっており、伝送クロック TCK などによりダウンカウントされてカウンタ値が“0”になると、カウンタマッチ信号を出力するようになっている。また、カウンタ 77 S, 77 R は、マルチプレクサ 74 S の出力信号でリセットされるようになっている。

10 【0064】マルチプレクサ 76 S, 76 R の出力信号は、SR フリップフロップ 79 の S, R 入力端子に夫々与えられている。RS フリップフロップ 79 の出力信号は、マルチプレクサ 80 に直接入力されていると共に、インバータゲート 81 を介して入力されている。そして、マルチプレクサ 80 の出力信号が、汎用出力信号として外部に出力される。尚、SR フリップフロップ 79 のセット側 S とリセット側 R とに同一のトリガ信号を選択した場合は、トリガ信号が発生する毎に出力レベルが反転するトグル動作となる。

20 【0065】図 17 は、汎用出力信号の出力例を示すタイミングチャートである。出力開始トリガとして FSTA を選択し、FSTA の出力タイミングでカウンタ 77 S, 77 R がリセットされると共にレジスタ 78 S, 78 R に設定されたデータがロードされカウンタが開始される。そして、カウンタ 77 S のカウンタマッチ信号が出力されると RS フリップフロップ 79 がセットされ、カウンタ 77 R のカウンタマッチ信号が出力されると RS フリップフロップ 79 がリセットされるようになる。そのセット、リセットの時間差によって汎用出力信号が出力されるようになっている。

30 【0066】次に、シリアル通信装置 11 を、具体的なプロトコル（TAB2）に適用した場合の例を図 18 乃至図 22 をも参照して説明する。TAB2（Telecontrol Audio Bus 2）は、（株）富士通テンによって仕様が定められたオーディオ用のシリアルバスである。図 18 は、システム全体の構成を示すものである。シリアル通信装置 11、CPU 12、DMAC 14 及びメモリ（RAM）100 は、例えば、カーナビゲーション装置を制御する ECU（Electronic Control Unit、マイクロコンピュータ）101 の一部として構成されている。そして、接続先デバイスであるカーオーディオ装置 102 は、TAB2 インターフェイス（I/F）部 103 及びオーディオ本体部 104 とで構成されている。

【0067】TAB2 で使用される通信制御信号には以下のものがあり、シリアル通信装置 11 の端子は、括弧内のものが割り当てられる。

TCK : 同期クロック（CLKOUT）

MDT : 送信データ（TX）

SDT : 受信データ（RX）

50 SYN : 同期初期化要求（UOUT1）

ANS : 通信制御に対するスレーブの応答 (UIN 3)

TRQ : バス使用権譲渡要求 (UIN 2)

そして、シリアル通信装置 11 は、マスタとして機能する。

【0068】<送信処理>図 19 は、CPU 12 がオーディオ装置 102 に対してデータを送信する場合のフローチャートであり、図 20 は送信のタイミングチャートである。図 19 の左側に示すものは CPU 12 側が行うソフトウェア (S/W) 的な処理であり、図 19 の右側に示すものはシリアル通信装置 11 側が行うハードウェア (H/W) 的な処理である。

【0069】CPU 12 は、まず、ステップ①において、(1) 送信を行うための初期化、(2) 送信データのチェックサムである FCC の計算及び (3) DMA 転送の初期設定を行う。以下により具体的な項目を示す。尚、以下の項目の設定は、送信制御部、受信制御部の各種レジスタに CPU 12 が書き込みを行うことで行うようになっている。

(1) 送信初期化

(SYN 出力設定)

・汎用出力 UOUT 1 より同期初期化要求信号 SYN を CPU 12 のレジスタ書き込みで行うことを設定。

(送受信初期設定)

・同期プロトコル、バスデータサイズ (8 ビット)、割り込み処理等の設定。

・バイト ACK 信号を汎用入力 UIN 3 (ANS) に割り当てる。

・同期クロック TCK として出力されるクロック信号に、内部のデータシフトクロック SCK を選択。

【0070】(汎用入力)

・汎用入力 UIN 1, 2 は S/W 設定。ハイレベル検出。“1” をセット。

・汎用入力 UIN 3 (ANS) は外部入力。立上がりエッジ検出。

(送信制御)

・TX_FSTA に送信許可信号 TE を選択。

・TX_BSTA に汎用入力 UIN 2 を選択。

・TX_FEND にデータサイズカウンタマッチ TX_DCNTM を選択。及び送信データサイズの設定。

・バイトサイズカウンタを TX_BSTA でセット。バイトサイズ N=8。

・応答信号 ANS の返信待ちタイムアウトをカウントするマルチカウンタ TX_MCNT の設定。

【0071】(受信制御) 尚、ここでの受信制御は、後述するようにオーディオ装置 102 側より 1 フレーム分のデータ受信の確認応答として送信される信号 ACK が受信データライン SdT によって送信されるのを、シリアル通信装置 11 が受信する必要があるからである。

・RX_FSTA, RX_BSTA に汎用入力 UIN 1 を選択。

・RX_FEND にデータサイズカウンタマッチ RX_DCNTM を選択。及び受信データサイズの設定 (8 ビット)。

・バイトサイズカウンタを RX_BSTA でセット。バイトサイズ N=8。

【0072】(汎用マッチ)

・汎用マッチ出力に、送信カウンタ TX_MCNT の出力を選択。

・汎用マッチ出力をエラー信号として出力。

(2) FCC の計算は、CPU 12 側で行われる。

(3) DMA 初期設定

・CPU 12 が、メモリ 100 に送信データ列を書込む。

・DMAC 14 のモード設定 (送信データサイズ、データ量、ソース及びデスティネーションアドレス) を、CPU 12 が DMAC 14 の内部レジスタに書き込んで行う。

・DMA 転送の使用設定 (転送要求 DREQ の出力許可)。

以上までが、ステップ①に相当する。

20 【0073】次のステップ②において、CPU 12 は、同期初期化要求信号 SYN をシリアル通信装置 11 に出力させるためレジスタに書き込みを行う。まず、UOUT 1 を “1” にセットした後、2ms ~ 5ms の経過後に “0” にセットするように書き込みを行う。すると、ステップ③においてシリアル通信装置 11 の UOUT 1 から図 20 に示すようにロウレベルのパルス信号 SYN が出力される。

30 【0074】続くステップ④では、CPU 12 がシリアル通信装置 11 の内部レジスタに書き込みを行うことにより送信許可信号 TE を発生させる。また、DMAC 14 のカウンタをセットする。すると、フレームスタート信号 TX_FSTA が出力され、送信ステートマシン 38 は “フレームスタート待ち” → “バイトスタート待ち” へと移行する。

【0075】また、送信バッファ 32 は初期状態で空であり、転送要求 DREQ の出力が許可されることでエンブティ信号 TBEMPT が DREQ として DMAC 14 に出力される。すると、DMAC 14 は、応答信号 DACK を通信装置 11 に返してメモリ 100 のソースアドレスから送信バッファ 32 のデスティネーションアドレスへデータを転送するので、エンブティ信号 TBEMPT はインアクティブとなる。

50 【0076】バイトスタート信号 TX_BSTA には汎用入力 UIN 2 が選択され (1) (送信初期化) の (汎用入力) において既に “1” がセットされているので、信号 TBEMPT までの論理積によってバイトスタート信号 TX_BSTA が出力される。すると、送信ステートマシン 38 は “バイトスタート待ち” → “送信中” へと移行する。送信バッファ 32 に転送された 8 ビットの送信データ (最初は “FH”) はシフトレジスタ 33 に転送され、1 ビ

ットずつシリアルに送信データMDTとして送信される。そして、送信バッファ32が空になるとエンブティ信号TBEMPT→転送要求信号DREQが発行され、DMAC14に次のデータ転送を促す。

【0077】データの送信中は、図3(f)に示す受信中信号RENGに相当する送信中信号TENGが出力されているので、各ビットの送信に合わせて同期クロックTCKとしてシフトクロックSCKが出力される。そして、バイトサイズカウンタでN=8に設定されているので、8ビットの送信が完了した時点でバイトエンド信号TX_BENDが出力される。

【0078】すると、送信ステートマシン38は“バイト受信終了処理”：ステップ⑥に移行し、カーオーディオ装置102のTAB2インターフェイス部103よりスレーブ応答ANSが返信されるのを2ms以内で待つ。その間にスレーブ応答ANSが返信されると(「YES」)ステップ⑦に移行して、フレームエンド信号TX_FENDが出力されたか否かを判定する。

【0079】また、ステップ⑥において、2ms以内にスレーブ応答ANSが返信されなければ、マルチカウンタTX_MCNTよりカウンタマッチ信号が出力され、CPU12に対して割込みを発生させる。この時、送信ステートマシン38は“バイト受信終了処理”から“エラー”に移行する。すると、CPU12は、ステップ⑧においてシリアル通信装置11内部の割込みレジスタの内容を参照し、割込み要因(TMCERR_INT)によりスレーブ応答タイムアウトであることを判断するとタイムアウトエラー処理を行ってからステップ②に戻る(送信ステートマシン38：“フレームスタート待ち”)。また、ステップ⑧ではDMAC14のカウンタをクリアする。

【0080】一方、ステップ⑦において、トリガ信号TX_FENDの出力条件にはデータサイズカウンタマッチTX_DCNTMが選択されており、設定された送信データサイズ分の送信が完了しなければTX_FENDは出力されない。そして、ステップ⑦で「NO」であればステップ⑤(送信ステートマシン38は“バイトスタート待ち”)に戻り、同時にトリガ信号TX_BSTAが発行されて次のバイトデータを送信する。

【0081】以上のようにして定められたデータサイズの送信が完了すると、ステップ⑦においてトリガ信号TX_FENDが発行され(「YES」)、CPU12に対して割込みを発生させる(TX_END_INT)。すると、CPU12は、ステップ⑨において受信初期化処理を以下のように行う。

- ・送信禁止：TE=0
- ・送信終了フラグクリア：TX_END_INT=0
- ・バイトACK信号無し。
- ・受信許可：RE=1

【0082】受信許可信号REがアクティブになると、汎用入力UIN1=1、との論理積でトリガ信号RX_FST 50

Aが発行され、ステップ“10”に移行する。また、トリガ信号RX_BSTAにも汎用入力UIN1が割り当てられているので、図20に示すように、トリガ信号RX_BSTAはステップ①の時点で既にアクティブとなっている。従って、受信ステートマシン30は直ちに“受信中”となる。

【0083】そして、図20に示すように、TAB2インターフェイス部103より送信された送信応答ACKのパターンデータ(8ビット)を受信データSDTとして受信すると、トリガ信号RX_BEND、RX_FENDが発行され、シリアル通信装置11はCPU12に受信終了割込みRX_END_INTを発行し、CPU12は受信終了処理(必要なレジスタ設定)を行う。以上が送信処理の一連の流れである。

【0084】<受信処理>次に、受信処理について図21及び図22をも参照して説明する。図21は、受信処理の手順を示すフローチャートである。

①初期設定

(SYN出力設定)：送信と同様。

(送受信初期設定)：送信と同様。

(汎用入力)

・汎用入力UIN1はS/W設定。ハイレベル検出。

“1”をセット。

・汎用入力UIN2は外部入力。ロウレベル検出。→TRQ検出用。

・汎用入力UIN3(ANS)は外部入力。立上がりエッジ検出。

・汎用入力UIN2にS/W設定用の“1”をセット(後で、ステップ⑤で切り換えて使用する)。

(送信制御)：送信と同様。

【0085】(受信制御)

・受信データサイズの設定(16ビット)以外は送信と同様。

・受信側マルチカウンタRX_MCNTを、TX_MCNTと同様に設定。

(パターンマッチ)

・シフトクロックSCKによりイネーブル信号SEの有効区間に受信データをサンプリング(NAQデータ)。

(汎用マッチ)

・送信側は送信と同様。

・受信側を送信側と同様に設定。

(DMA初期設定)

以上までが、ステップ①に相当する。

【0086】次のステップ②において、CPU12は、シリアル通信装置11の割込み要因レジスタをポーリングして、TAB2インターフェイス部103がバス使用権譲渡要求信号TRQをロウレベルにドライブすることを検出するまで待機する。即ち、割込み要因レジスタには、汎用入力2の検出割込み要求状態を示すビットがあり、①の設定により、UIN2がロウレベル(TRQ=

L) になると上記ビットに“1”がセットされるようになっている(尚、この要因による割込みは、設定によりマスクされている。)。そして、CPU12がTRQ=Lを検出すると(「YES」)、送信処理と同等にして同期初期化要求信号SYNをシリアル通信装置11に出力させる(ステップ③、④)。それから、CPU12は、ENQの送信設定を行なう(ステップ⑤)。

【0087】ステップ⑤では、送信バッファ32に送信データENQをセットし、汎用入力UIN2をS/W設定、ハイレベル検出に切り換える。そして、送信許可TE=1とすると、①の(汎用入力)において汎用入力UIN2にS/W設定用の“1”がセットされているので、トリガ信号TX_FSTA, TX_BSTAが出力される。すると、シリアル通信装置11によって、データENQがTAB2インターフェイス部103側に送信される(ステップ⑥)。

【0088】続いては、送信処理と同様に、TAB2インターフェイス部103よりスレーブ応答ANSが返信されるのを2ms以内に待つ(ステップ⑦)。その間にスレーブ応答ANSが返信されると(「YES」)、CPU12に対して送信終了割込みTX_END_INTを発生させる。

【0089】すると、CPU12は、FH/NAQの受信設定を行なう(ステップ⑧)。即ち、送信禁止:TE=0、送信終了フラグクリア、受信許可:RE=1を設定する。そして、シリアル通信装置11はデータの受信を行ない(ステップ⑨)、スレーブ応答ANSの返信を待つ(ステップ“10”)。スレーブ側のTAB2インターフェイス部103は、マスタ側のシリアル通信装置11に送信すべきデータがある場合は“FH”を送信して応答ANSを返す。また、送信すべきデータがない場合は“NAQ”を送信し、応答ANSは返さない。

【0090】従って、ステップ“10”で応答ANSを受信すると(「YES」)、シリアル通信装置11は、続いてFDLを受信する(ステップ“12”)。また、応答ANSを受信しなければ(「NO」)、シリアル通信装置11は、CPU12に対して受信汎用マッチ割込み(RMC_INT)を発生させる。CPU12は、その割込みを受けて受信データを読み出し、データがNAQか否かを判定する。そして、受信データがNAQであれば(「YES」)ステップ⑩に戻り、NAQでなければタイムアウトエラー処理を行なう(ステップ“19”)。

【0091】ステップ“12”でFDLを受信すると、シリアル通信装置11は、ステップ“13”に移行してスレーブ応答ANSの返信を待ち、ANSの返信があると(「YES」)CPU12に対して受信終了割込みRX_END_INTを発生させる。すると、CPU12は、受信データFH, FDLについてヘッダ処理を行い(ステップ“14”)、続いて、DATA~FCCの受信設定を行なう(ステップ“15”)。

【0092】また、ステップ“13”において、2ms以内にスレーブ応答ANSが返信されなければマルチカウンタRX_MCNTよりカウンタマッチ信号が出力され、CPU12に対して割込みRMCERR_INTを発生させる。すると、CPU12は、ステップ“19”に移行して送信処理の場合と同様にタイムアウトエラー処理を行ってからステップ③に戻る。

【0093】ステップ“15”では、一旦受信禁止とし(RE=0)、受信終了フラグをクリアしてからDATA~FCCの受信データサイズをセットする。更に、DMAC14に対する転送要求DREQの出力を許可すると共に、再び受信を許可する(RE=1)。すると、トリガ信号RX_FSTA及びRX_BSTAが出力される。

【0094】続くステップ“16”において、シリアル通信装置11は、TAB2インターフェイス部103より送信されたデータが受信バッファ23に格納されてバッファ23がフル状態になると、フル信号RBFULを転送要求信号DREQとしてDMAC14に出力する。すると、DMAC14は、受信バッファ23から読み出したデータをメモリ100のデスティネーションアドレスへ転送する。

【0095】そして、シリアル通信装置11は、1バイトのデータを受信するとスレーブ応答ANSの返信を待ち(ステップ“17”)、2ms以内に返信があれば(「YES」)次のステップ“18”に移行してトリガ信号RX_FENDが出力されたか否かを判定する。また、ステップ“17”において、2ms以内にスレーブ応答ANSが返信されなければ(「NO」)、ステップ“13”と同様にステップ“19”に移行する。

【0096】ステップ“18”においては、設定された受信データサイズ分の受信が完了しなければRX_FENDは出力されない。そして、ステップ“18”で「NO」であればステップ“16”に戻り、同時にトリガ信号RX_BSTAが発行されて次のバイトデータを受信する。また、全てのデータの受信が完了した場合は(「YES」)、CPU12に対して受信終了割込みRX_END_INTを発生させる。すると、CPU12は、ACK送信設定を行なう(ステップ“20”)。ACK送信設定は、受信を禁止し(RE=0)受信終了フラグをクリアしてから送信バッファ32に送信データACKをセットし、送信を許可する(TE=1)。

【0097】すると、シリアル通信装置11は、ACKデータをTAB2インターフェイス部103側に送信して(ステップ“21”)、CPU12に対して送信終了割込みTX_END_INTを発生させる。

【0098】以上のように本実施例によれば、シリアル通信装置11を、トリガ信号発生手段によってFSTA, FEND, BSTA, BENDの各トリガ信号を発生させるように構成し、これらのトリガ信号の発生に基づいて送信ステートマシン38、受信ステートマシン3

0の状態を遷移させ、通信制御信号の発生状態に応じて必要な出力信号をTAB2インターフェイス部103に出力させるように構成した。

【0099】従って、カーナビゲーション装置のCPU12が、カーオーディオ装置102のTAB2インターフェイス部103とシリアル通信を行う場合に、設定に応じて対処することができる。そして、従来はサブプロセッサ5が行っていた通信処理をシリアル通信装置11のハードウェアによって代行することになり、図58に示す従来構成では、汎用ポート3、シリアルポート4及びサブCPU5によって実現されていた機能をシリアル通信装置11に置き換えることができる。従って、サブプロセッサ5を用いずともCPU12に対して発生させる割り込みを削減することが可能となり、CPU12の処理効率を向上させることができる。また、ECU101を低コストで構成することが可能となる。

【0100】また、各プロトコルに対応してホストに実行させるプログラムを、シリアル通信装置11に用意されているレジスタの設定に合わせて作成することができるので、プログラミングの手法をある程度標準化させることが可能となり、プログラミング作業をより容易に行なうことができる。

【0101】更に、ECU101を、シリアル通信装置11、CPU12、DMAC14、及びメモリ100で構成したので、シリアル通信装置11は、CPU12に対してバイトデータの送信、受信毎に割り込みを発生させる必要がなく、送信、受信処理の終了時に割り込みを発生させれば良いので、CPU12の割り込み処理負担を大きく軽減することができる。

【0102】(第2実施例) 図23乃至図28は本発明の第2実施例を示すものである。第2実施例では、シリアル通信装置11を、RS-232C(無手順)に適用した場合であり、図23は、シリアル通信装置11をRS-232Cモジュールとして機能させる場合における入出力信号の割当状態を示すものである。

CLKIN : 入力クロック (CLKIN)
RXD : 受信データ (RX)
RTS : 送信要求信号 (UOUT2)
CLKOUT : 出力クロック (CLKOUT)
TXD : 送信データ (TX)
CTS : 送信許可信号 (UIN2)

尚、RTS、CTSは負論理である。

【0103】<送信処理> 図24は送信処理のフローチャート、図25は同タイミングチャートを示す。また、図26は、送信ステートマシン28の構成を図7のレベルよりも詳細に示すものである。図26において、①“フレームスタート待ち”から②“バイトスタート待ち”に移行する場合は、条件(TX_FSTA ¬ FLSYNC)が成り立つ場合であり、条件(TX_FSTA & FLSYNC)が成り立つ場合は、①a“フラグ挿入”のステートを経由す

る。

【0104】また、②“バイトスタート待ち”から③“送信中”に移行する場合は、条件(TX_BSTA ¬ STASTP)が成り立つ場合であり、条件(TX_BSTA & STASTP)が成り立つ場合は、②a“スタートビット挿入”のステートを経由する。更に、③“送信中”において条件(BIT_STF & TX_MCNTM)が成り立つ場合は、③a“0ビット挿入”のステートを経由して③に戻る。

【0105】また、③“送信中”から直接④“バイト受信終了”に移行するのは、条件(TX_BEND ¬ PE ¬ STASTP)が成り立つ場合であり、条件(TX_BEND & PE)が成り立つ場合は、③b“パリティ挿入”のステートに移行する。③bにおいて条件STASTPが成り立つ場合は③c“ストップビット挿入”のステートを経由して④に移行し、条件notSTASTPが成り立つ場合は直接④に移行する。加えて、③において条件(TX_BEND ¬ PE & STASTP)が成り立つ場合は③bに移行する。但し、第2実施例では、①a“フラグ挿入”及び③a“0ビット挿入”のステートに移行するケースはなく、これらは後述の第3実施例において利用される。

【0106】再び図24を参照して、先ず、ステップ①では、CPU12は以下の設定を行なう。

(送信初期化)

- ・非同期プロトコル、全二重通信の設定。
- ・パリティビット(PE)、スタート・ストップビット(STASTP)の設定。
- (汎用入力設定)
- ・汎用入力2(UIN2)を外部入力に設定(CTS)。ロウレベル検出。

(送信制御)

- ・TX_FSTAに汎用入力2(UIN2)を選択。
- ・TX_BSTAに(UIN2 ¬ TBEMPT)を選択。
- ・TX_FENDにデータサイズカウンタマッチTX_DCNTMを選択。及び送信データサイズの設定。
- ・バイトサイズカウンタをTX_BSTAでセット。バイトサイズN=8。

(DMA初期設定)

以上までが、ステップ①に相当する。

【0107】次のステップ②では、CPU12が送信許可TE=1を行い、送信を開始させる。すると、シリアル通信装置11は、汎用入力UIN2がロウレベルにドライブされるまで待機し(ステップ③)、ロウレベルにドライブされると(「YES」)トリガ信号TX_FSTAを発生させてステップ④に移行する。DMAC14により最初の送信データが送信バッファ32に転送されると、エンプティ信号TBEMPTはインアクティブとなりトリガ信号TX_BSTAが発生される。そして、送信データTXDはシフトレジスタ33よりシリアルに送信されるが、この時、条件STASTPと条件PEとが成り立っているので、データの送信に先立ってビット挿入部62によりスタートビ

ットの挿入が行われる。

【0108】8ビットのバイトデータが送信されるとトリガ信号TX_BEND が発生されるが、付帯条件 (&PE&STASTP) によって、バイトデータ送信の送信後にビット挿入部62によりパリティビット、ストップビットの付加が行われる。そして、シリアル通信装置11は、ステップ⑤においてデータサイズカウンタによる送信終了判定を行い、設定されたデータサイズの送信が完了していなければステップ③に戻り、送信が完了した場合はCPU12に対して送信終了割込みTX_END_INTを発生させる。

【0109】<受信処理>図27は受信処理のフローチャート、図28は同タイミングチャートを示す。先ず、ステップ①の(受信初期化)は、送信処理と全く同様に行われる。

(汎用入力設定)

・汎用入力1(UIN1)をS/Wに設定。ハイレベル検出。“1”をセット。

(汎用出力設定)

汎用出力2(UOUT2)から送信要求信号RTSを出力するための設定を行なう。

・出力開始トリガ、出力終了トリガに受信バッファフル信号RBFUL を選択。

【0110】(受信制御)

・RX_FSTA に汎用入力UIN1を選択。

・RX_BSTA にパターンマッチ1を選択。

・RX_FEND にデータサイズカウンタマッチRX_DCNTMを選択。及び受信データサイズの設定(8ビット)。RX_BSTA でカウンタセット。

・バイトデータサイズの設定(8ビット)。

(パターンマッチ)

・伝送クロックTCKによりクロック毎サンプリング(スタートビット検出)。第0～第6ビットはマスクする。

・パリティエラー検出。

(DMA初期設定)：第1実施例の受信処理と同様。

【0111】次に、CPU12は、受信許可RE=1を設定して受信を開始させる(ステップ②)。すると、シリアル通信装置11は、パターンマッチ1によってスタートビット“0”の受信を検出するまで待機し(ステップ③)、受信を検出すると(「YES」)RX_BSTA を発生させてバイトデータを受信する(ステップ④)。尚、実際の受信開始は接続先デバイスによるデータ送信タイミングで決まるため、RX_BSTA の発生は、RX_FSTA の発生よりも後になるのが通常である。ステップ④において受信バッファ23がフルとなり、バッファフル信号RBFUL=1となると、DMAC14に転送要求DREQを出力する。

【0112】また、ステップ⑨～“11”は、他のステップとは独立のシリアル通信装置11のH/W処理ルーチンである。即ち、受信バッファ23がフルでなければ

(ステップ⑨, 「NO」)(汎用出力設定)に基づき送信要求信号RTSをアクティブ“0”にし(ステップ“10”)、受信バッファ23がフルであれば(ステップ⑨, 「YES」)送信要求信号RTSをインアクティブ“1”にする(ステップ“11”)。

【0113】ステップ⑤, ⑥では、パリティエラー、フレーミングエラー(ストップビット検出無し)の検出を行なう。エラーが検出されると(「YES」)、CPU12に対して割込みPCERR_INT, FERR_INTを発生させ、エラー処理を実行させる(ステップ⑧)。ステップ⑧でエラー処理を行った後はステップ②に移行する。そして、ステップ⑤, ⑥でエラーが検出されなければ(「NO」)、データサイズカウンタによる受信終了判定を行い(ステップ⑦)、設定されたデータサイズの受信が完了していなければステップ③に戻り、受信が完了した場合はCPU12に対して受信終了割込みRX_END_INTを発生させる。

【0114】以上のように第2実施例によれば、送信データ列にビットデータの挿入を行うためのビット挿入部62を備え、送信ステートマシン38を、バイトスタート待ちステートと送信中ステートとの間においてスタートビット挿入ステートに遷移可能に構成し、また、通信中ステートとバイト通信終了ステートとの間にストップビット挿入ステートに遷移可能に構成した。

【0115】更に、通信中ステートとバイト通信終了ステートとの間において、送信データにパリティビットを挿入するためのパリティビット挿入ステートに遷移可能に構成した。従って、RS-232Cのように、送信データの最初と最後にスタートビット、ストップビットやエラー検出用のパリティビットを場合によって付加する必要があるプロトコルにも対応することができる。

【0116】(第3実施例)図29乃至図33は本発明の第3実施例を示すものである。第3実施例では、シリアル通信装置11を、HDL C(High Level Data Link Control procedure)に適用した場合であり、図29は、シリアル通信装置11をHDL Cモジュールとして機能させる場合における入出力信号の割当状態を示すものである。

CLKIN : 非同期クロック(CLKIN)

TX : 送信データ(TX)

RX : 受信データ(RX)

CLKOUT : 非同期クロック(CLKOUT)

【0117】<送信処理>図30は送信処理のフローチャート、図31は同タイミングチャートを示す。先ず、ステップ①では、CPU12は以下の設定を行なう。

(送信初期化)

・非同期プロトコル、全二重通信の設定。

・ビットスタッフィング(BIT_STF), フラグ同期(FL SYNC)ありの設定。

(汎用入力設定)

10

20

30

40

50

・汎用入力2 (UIN2) をS/W設定。ロウレベル検出。“1”をセット。

(送信制御)

- ・TX_FSTA にTEを選択。
- ・TX_BSTA に(UIN2 & not TBEMPT)を選択。
- ・TX_FEND にデータサイズカウンタマッチTX_DCNTMを選択。
- ・送信マルチカウンタ設定(ビットスタッフィングに利用される)。TCKカウント, TX=0でセット、カウンタサイズ=5。
- ・バイトサイズカウンタTX_BSTA でセット。バイトサイズN=8。
- ・フラグシーケンスデータ0x7eをレジスタにセット。

【0118】(エラー検出)

・CRC(Cyclic Redundancy Check)によるエラー検出設定。

(DMA初期設定)

- ・メモリ100に送信データ列(フラグ, CRCを除く)を書込む。
- ・DMAC14のモード設定。
- ・DMA転送の使用設定。

以上までが、ステップ①に相当する。

【0119】次のステップ②では、CPU12が送信データサイズを設定して(CRCを除く)送信許可TE=1を行い送信を開始させる。この時点でトリガ信号TX_FSTAの発生条件が成立するが、シリアル通信装置11は、付帯条件(&FLSYNC)によって先ず内部レジスタにセットされたフラグシーケンスデータ(0x7e)を送信するためのタイミングに利用する(ステップ③)。それから、内部的にトリガ信号TX_FSTAを出力すると、第1, 第2実施例と同様に、DMAC14より送信バッファ32に転送されるバイトデータ(アドレス, 制御, 情報)を順次送信する(ステップ④, ⑤)。

【0120】尚、この場合、ビットスタッフィング(BIT_STF)がアクティブとなっているので、シリアル通信装置11はビット挿入部62により送信データに必要な応じて“0”ビット挿入を行う。即ち、送信マルチカウンタをTX=0でカウンタサイズ=5をセットしTCKをカウントする。そして、TX=1の状態を5回カウントした場合は、フラグシーケンスと同じビットパターンが発生していることになる。従って、その時点で送信データに“0”ビット挿入を行うことで、フラグシーケンスと同一のビットパターンを送信することを回避する。

【0121】そして、シリアル通信装置11は、ステップ⑤においてデータサイズカウンタによる送信終了判定を行い、設定されたデータサイズの送信が完了すると(<Y>)ステップ⑥においてCRCデータを送信する。それから、TX_BENDと共にTX_FENDを発生させてフラグシーケンスデータを送信し(ステップ⑦)、CPU12に

対して送信終了割込みTX_END_INTを発生させる。

【0122】<受信処理>図32は受信処理のフローチャート、図33は同タイミングチャートを示す。先ず、ステップ①では、CPU12は以下の設定を行なう。

(受信初期化)：送信と同じ。

(汎用入力設定)

・汎用入力1 (UIN1) をS/W設定。ロウレベル検出。“1”をセット。

(受信制御)

- 10 ・RX_FSTA に(RE & パターンマッチ1)を選択。
- ・RX_BSTA に汎用入力1 (UIN1)を選択。
- ・RX_FEND にパターンマッチ1を選択。
- ・受信データサイズを最大にセット。
- ・バイトサイズカウンタRX_BSTA でセット。バイトサイズN=8。
- ・受信マルチカウンタ設定(スタッフィングされた“0”ビット削除に使用)。TCKカウント, RX=0でセット、カウンタサイズ=5。
- ・フラグシーケンスデータ0x7eをレジスタにセット。

20

【0123】(パターンマッチ)

- ・伝送クロックTCKで、クロック毎にサンプリング。
- ・パターンマッチ1のデータをフラグ検出用(0x7e)に設定。
- ・パターンマッチ2のデータをアボート検出用(0xff)に設定。第8ビットをマスク。

(エラー検出)

- 30 ・CRC(Cyclic Redundancy Check)によるエラー検出設定。
- ・パターンマッチ2をエラー検出用に選択。

(DMA初期設定)

以上までが、ステップ①に相当する。

【0124】次のステップ②では、CPU12が受信許可RE=1を設定し受信を開始させる。すると、シリアル通信装置11は、フラグシーケンスデータ(0x7e)の受信を待つ(ステップ③, ④)。フラグシーケンスを受信した場合は、シリアル通信装置11内部でパターンマッチ1が出力され(ステップ④, <Y>)、受信許可RE=1とのAND条件によってトリガ信号RX_FSTAが発生されると共に、トリガ信号RX_BSTAも発生される。

40

【0125】続くステップ⑤～⑦は、フラグシーケンスに続いて送信される同期(制御)フラグを受信する。ステップ⑤で受信した8ビットデータがアボートパターンである場合は(ステップ⑥, 「YES」)、CPU12に受信マッチエラー割り込みRMCERR_INTを発生させる。すると、CPU12は、そのエラーが必要なデータの受信後に発生したものか否かを判断し(ステップ“13”)、必要なデータの受信後であれば(「YES」)エラー要因をクリアして受信停止RE=0とし、処理を

50

終了する。また、必要なデータの受信中に発生したエラーであれば（「NO」）、ステップ“14”に移行してエラー処理を行った後、ステップ②に戻る。

【0126】一方、ステップ⑤で受信した8ビットデータがフラグである場合は（ステップ⑦、「YES」）ステップ⑤に戻り、前記データが受信データ（情報）であれば（ステップ⑦、「NO」）、データを8ビット単位で受信し、受信バッファフル信号RBFUL = 1であればDMA14に転送要求DREQを出力する（ステップ⑧）。

【0127】この場合、シリアル通信装置11は、受信データ中にスタフィングされている“0”ビットを検出すると、その“0”ビットの削除を行う。即ち、受信マルチカウンタをRX=0でカウントサイズ=5をセットしTCKをカウントする。そして、RX=1の状態を5回カウントした場合はその次のビットに“0”ビットが挿入されているので、その“0”ビットを削除する。

【0128】そして、情報データの受信中においてもアボートパターンを検出すると（ステップ⑨、「YES」）ステップ“13”に移行する。そして、ステップ“10”において、パターンマッチ1が出力されるか否かによってフラグシーケンスを受信したか否かを判断し、受信しなければ（「NO」）ステップ⑧に戻り、受信した場合は（「YES」）CPU12に受信終了割込みRX_END_INTを発生させる。すると、CPU12は、その時点までに受信したデータの内容から受信終了と判断すると（「YES」）、その時点で強制的に受信停止RE=0とし（ステップ“12”）処理を終了する。

【0129】以上のように第3実施例によれば、シリアル通信装置11の送信ステートマシン38を、フレームスタート待ちステートとバイトスタート待ちステートとの間において、内部レジスタの設定に応じてフラグデータ挿入ステートに遷移可能に構成したので、HDL Cのように、送信の開始時と終了時にフラグシーケンスを送信する必要があるプロトコルにも対応することが可能となる。また、パターンマッチによりフラグシーケンスを検出することで、シリアル通信装置11を、フラグ同期型のプロトコルであるHDL Cのモジュールとして使用することができる。

【0130】（第4実施例）図34乃至図38は本発明の第4実施例を示すものである。第4実施例では、シリアル通信装置11を、IIC(Inter Integrated Circuit)バスに適用した場合であり、図34は、シリアル通信装置11をIICモジュール（マスタ、スレーブ）として機能させる場合における入出力信号の割当状態を示すものである。

SCL : 同期クロック（CLKOUT, CLKIN）

SDA : 通信データ（TX, RX）

【0131】＜送信処理＞図35は送信処理のフローチ

ャート、図36は同タイミングチャートを示す。先ず、ステップ①では、CPU12は以下の設定を行なう。

（送信初期化）IICプロトコルでは、マスタ側が送信開始時にスタートコンディション（SCLがハイレベルの場合にSDAを立下げる）をスレーブ側に出力することでデータの送受信が開始される。そのスタートコンディションを出力するための設定を行なう。

- ・同期プロトコル。半二重通信。TXの初期値=1。
- ・スレーブからのBACK信号を、データラインTXで

10 受信。

- ・フレーム開始同期信号として“0”出力。データラインTXで送信。

- ・フレーム終了同期信号無し。

- ・CLKOUTにシフトクロックSCKを選択。

- ・DREQ出力禁止。

【0132】（汎用入力）・汎用入力2（UIN2）をS/W設定。ロウレベル検出。“1”をセット。

（送信制御）

- ・TX_FSTA にTEを選択。

20 TX_BSTA に汎用入力2（UIN2）を選択。

- ・TX_FEND にデータサイズカウンタを選択。

- ・バイトサイズカウンタをTX_BSTA でセット、バイトサイズN=8。

（DMA初期設定）

（送信許可）

- ・送信データサイズ8ビット。送信許可TE=1（TX_FSTA 発生）

以上までが、ステップ①に相当する。

【0133】次のステップ②では、CPU12が送信バッファ32にアドレス及びR/Wのデータを書込んでセットすると、送信バッファ32のエンプティ信号がインアクティブになり、トリガ信号TX_BSTA が発生する。そして、ステップ③において、シリアル通信装置11がSDA（TX）のレベルを“0”にすることでスタートコンディションを出力し、続いて、送信バッファ32にセットされたアドレス及びR/Wのデータを送信する（ステップ④）。

【0134】そして、シリアル通信装置11は、データラインTXがスレーブによってロウレベルにドライブされることでスレーブ応答ACKを受信したか否かを判断し（ステップ⑤）、受信した場合は（「YES」）CPU12に対して送信終了割込み（TX_END_INT）を発生させる。また、スレーブ応答ACKを受信しなかった場合は（「NO」）、CPU12に対して送信フレーミングエラー割込み（TFERR_INT）を発生させる。すると、CPU12はエラー処理（ステップ“10”）を行ってからステップ②に戻る。

【0135】また、スレーブ側がSCLをロウレベルにドライブすることでビジーステータスを示した場合、シリアル通信装置11は、そのビジーステータスが解除さ

れるまで次の送信処理を行わない。

【0136】CPU12は、送信終了割込みを受け付けると、送信初期化（ステップ⑥）を行う。即ち、一旦送信を禁止し（TE=0）、フレーム開始同期信号を“無し”に、フレーム同期終了信号として“0”出力及び送信データサイズを設定する。そして、DMAC14にDREQの出力を許可すると、DMAC14にカウンタの設定を行ない送信を許可（TE=1）する（ステップ⑦）。

【0137】続くステップ⑧、⑨、“11”において、シリアル通信装置11は、バイトデータの送信を行う。ステップ⑨においてスレープ応答ACKが返らなかった場合は、ステップ“10”に移行する。また、ステップ“11”においてデータサイズカウンタのマッチ出力(TX_FEND)が発生した場合は（「YES」）、ストップコンディション（SCLがハイレベルの場合にSDAを立上げる）をスレープ側に出力することで（ステップ“12”）、CPU12に対して送信終了割込み(TX_END_INT)を発生させる。

【0138】＜受信処理＞図37は受信処理のフローチャート、図38は同タイミングチャートを示す。ステップ⑩の送受信初期化は、送信処理の場合と略同様である。

（汎用入力設定）

・汎用入力1, 2（UIN1, 2）をS/W設定。ロウレベル検出。夫々“1”をセット。

（送信制御）

・TX_FSTA にTEを選択。
・TX_BSTA に汎用入力2（UIN2）を選択。
・TX_FEND にデータサイズカウンタを選択。
・バイトサイズカウンタをTX_BSTA でセット、バイトサイズN=8。

【0139】（受信制御）

・RX_FSTA に汎用入力1（UIN1）を選択。
・RX_BSTA に汎用入力2（UIN2）を選択。
・RX_FEND にデータサイズカウンタを選択。
・バイトサイズカウンタをRX_BSTA でセット、バイトサイズN=8。

（DMA初期設定）

・DMAC14のモード設定。DMA転送の使用設定。
（送信許可）
・送信データサイズ8ビット。
・送信許可TE=1

以上までが、ステップ⑩に相当する。続くステップ⑪～⑬は、送信処理と同様に行われる。また、ステップ⑬におけるエラー処理も、ステップ“10”と同様である。

【0140】CPU12は、シリアル通信装置11による送信終了割込みを受け付けると、受信初期化（ステップ⑪）を行う。ここでは、送信処理のステップ⑥と同様の設定を行なうと共に、データの受信に応じてSDAラ

インに出力するバイトACK信号の出力設定を行なう。即ち、バイトACK信号をロウレベル“0”として出力すると共に、最終バイトの受信に対応するACK信号はストップコンディションを出力することで不要となるため、ACK信号の出力をマスクするように設定する。以降のステップ⑭～“12”は、送信処理のステップ⑦～“12”を受信について行なうものである。

【0141】以上のように第4実施例によれば、シリアル通信装置11をIICバスのモジュールとして機能させるので、同一データラインでアドレス及びデータやACK信号を送信するようなプロトコルにも対応することができる。

【0142】（第5実施例）図39乃至図44は本発明の第5実施例を示すものである。第5実施例では、シリアル通信装置11を、IIS（Inter IC Sound）バスに適用した場合であり、図39は、シリアル通信装置11をIISモジュール（（a）クロックマスタ、（b）クロックスレーブ）として機能させる場合における入出力信号の割当状態を示すものである。

SCK : 同期クロック（CLKOUT, CLKIN）

WS : L/Rクロック（UOUT1, UIN1 or 2）

SDI : シリアルデータ（TX）

SDO : シリアルデータ（RX）

【0143】＜送信処理＞図40は送信処理のフローチャート、図41は同タイミングチャート（クロックマスタの場合）を示す。先ず、ステップ⑪では、CPU12は以下の設定を行なう。

（送信初期化）

・同期プロトコル、全二重通信。TXの初期値=1。MSBファースト。
・音声データ、前詰め送信。
・送受信時、自身（マスタ）がクロックマスタとなる設定。

（汎用入力設定）

・汎用入力2（UIN2）をS/W設定。ロウレベル検出。“1”をセット。

（汎用出力設定）

・汎用出力1（UOUT1）のセットトリガ、リセットトリガに送信マルチカウンタ出力を設定。S-Rフリップフロップ出力。

【0144】（送信制御）

・TX_FSTA にマルチカウンタ出力を選択。
・TX_BSTA に汎用入力2（UIN2）を選択。
・TX_FEND にデータサイズカウンタを選択。データサイズ16ビット。TX_FSTA でカウンタセット。
・バイトサイズ、N=16。

（送信マルチカウンタTX_MCNT 設定）

・TCKカウント、TE=1でセット。カウントサイズ

0 x 1 f

(DMA初期設定)

以上までが、ステップ①に相当する。

【0145】次のステップ②では、CPU12は、DMAC14にカウンタ設定を行ない、送信許可TE=1をセットする。すると、送信マルチカウンタにカウンタ値がセットされカウントが開始される(ステップ③)。次のステップ④において送信マルチカウンタのカウンタマッチ出力が検出されると(「YES」、TX_FSTA 発生)、それをトリガとしてステップ⑤に移行し、バイトデータの送信が開始される。即ち、WSがロウレベルの期間はLチャンネル用の16ビットデータが送信され、ハイレベルの期間はRチャンネル用の16ビットデータが送信される。

【0146】尚、送信マルチカウンタのカウンタマッチ出力は、UOUT1のフリップフロップ出力をトグルさせるので、その出力毎にL/RクロックWSのレベルは遷移する。そして、ステップ⑥では、送信データサイズカウンタの出力に基づいてTX_FEND を発生させる。

【0147】また、シリアル通信装置11がクロックスレーブとなる場合は、以下のようになる。先ず、図41におけるステップ③を“WS信号検出開始”に、ステップ④を“WS検出?”に置き換える。そして、ステップ①の(送信初期化)では、スレーブがクロックマスタとなる設定を行なう部分だけが相違する。

(汎用入力設定)

- ・汎用入力2(UIN2)を外部入力に設定(WS)。
- 立下がりエッジ検出(又は立上がりでも可)。
- ・汎用入力3(UIN3)をS/W設定。ロウレベル検出。“1”をセット。

(汎用出力設定)：不要。

【0148】(送信制御)

- ・TX_FSTA に汎用入力2(¬TBEMPT)選択。
- ・TX_BSTA に汎用入力3(UIN3)を選択。
- (送信マルチカウンタTX_MCNT 設定)：不要。
- ・TCKカウント、TE=1でセット。カウントサイズ0 x 1 f

(DMA初期設定)以上までが、ステップ①に相当する。そして、CPU12は、ステップ②を同様に実行すると、シリアル通信装置11はWS信号の(レベル変化)の検出を開始し(ステップ③)、WS信号を検出すると(「YES」)ステップ⑤に移行する。その他はクロックマスタの場合と同様である。

【0149】<受信処理>図42は受信処理のフローチャートを示す。ステップ①の送受信初期化は、送信処理の場合と同様である。尚、受信時のタイミングチャートは送信時と同様となるので、図示を省略する。

(汎用入力設定)

- ・汎用入力2(UIN2)をS/W設定。ロウレベル検出。“1”をセット。

(汎用出力設定)

- ・汎用出力1(UOUT1)のセットトリガ、リセットトリガに受信マルチカウンタ出力を設定。S-Rフリップフロップ出力。

【0150】(受信制御)

- ・RX_FSTA にマルチカウンタを選択。
- ・RX_BSTA に汎用入力1(UIN1)を選択。
- ・RX_FEND にデータサイズカウンタを選択。
- ・バイトサイズカウンタをRX_BSTA でセット、バイトデータサイズN=16。

(受信マルチカウンタRX_MCNT 設定)

- ・TCKカウント、RE=1でセット。カウントサイズ0 x 1 f。

(DMA初期設定)以上までが、ステップ①に相当する。そして、以降のステップ②～⑥は、送信処理を受信に置き換えたものとして行なう。但し、ステップ⑥において「YES」と判断した場合はステップ③に戻る。そして、CPU12は、DMAC14からのDMA終了割込みを受けると、データの受信を終了する(ステップ⑦)。

【0151】また、シリアル通信装置11がクロックスレーブとなる場合は、上記受信のフローチャートを、送信処理においてクロックスレーブとなる場合と同様に置き換えたものとなる。

【0152】以上のように第5実施例によれば、シリアル通信装置11をIISバスのモジュールとして機能させるので、L/RクロックWSのような通信制御信号によってデータの種別を判別するようなプロトコルにも対応することができる。

【0153】(第6実施例)図43乃至図47は本発明の第6実施例を示すものである。第6実施例は、シリアル通信装置11をHIT64に適用した場合であり、図43は、シリアル通信装置11をHIT64モジュールとして機能させる場合における入出力信号の割当状態を示すものである。

TX : 送信データ(TX)
RX : 受信データ(RX)
HBREQ : スレーブからのポーリング要求(UIN3)

【0154】<送信処理>図44は送信処理のフローチャート、図45は、シリアル通信装置11と接続先デバイスとの間における通信シーケンス((a)は制御フレーム、(b)はデータフレーム)を示す。先ず、ステップ①では、CPU12は以下の設定を行なう。

(送信初期化)

- ・非同期プロトコル、全二重通信。TXの初期値=1。
- ・スタート/ストップビット、夫々1ビット

(汎用入力設定)

- ・汎用入力1, 2(UIN1, 2)をS/W設定。ハイレベル検出。夫々“1”をセット。

【0155】・汎用入力3を外部入力設定。ハイレベル検出。

(送信制御)

- ・TX_FSTA にTEを選択。
- ・TX_BSTA に汎用入力2 (UIN2) を選択。
- ・TX_FEND にデータサイズカウンタを選択。データサイズ任意。
- ・バイトサイズカウンタをTX_FSTA でセット。バイトサイズN=8。

【0156】(受信制御)

- ・RX_FSTA に汎用入力1 (UIN1) を選択。
- ・RX_BSTA にパターンマッチ1を選択。
- ・RX_FEND にデータサイズカウンタを選択。受信データサイズ任意。
- ・バイトサイズカウンタをRX_BSTA でセット。バイトサイズN=8。

(パターンマッチ設定)

- ・伝送クロックTCKによりクロック毎サンプリング(スタートビット検出)。第0～第6ビットはマスクする。

(タイムアウト設定)

- ・送信フレーム間、受信フレーム間のタイムアウト時間を40msに設定。

(DMA初期設定)

以上までが、ステップ①に相当する。

【0157】そして、CPU12は、ステップ②でDMAC14のカウンタを設定すると共に送信を許可し(TE=1)、シリアル通信装置11はバイトデータの送信を開始する(ステップ③)。そして、データサイズカウンタにより送信終了と判断すると(ステップ④, 「YES」)、CPU12に対して送信終了割り込みを発生させてステップ⑤に移行する。尚、送信中にステップ“15”において送信フレームタイムアウトが発生すると(「YES」)、CPU12に対してタイムアウトエラー割り込みTX_FTOERR_INTを発生させる。すると、CPU12は、ステップ“17”においてエラー処理を行う。

【0158】CPU12は、ステップ⑤において送信を一旦停止させる(TE=0)。そして、送信したデータがMDHであれば(「YES」)データフレームの送信中であり、10msの間隔をとってから(ステップ⑥)ステップ②に移行する。また、送信したデータがMDHでなければ(ステップ⑤, 「NO」)、制御フレームENQの問い合わせ(データあり?)に対するスレーブ応答ACKを受信するため、受信フレームタイムアウトを許可してから(ステップ⑦)受信を許可する(RE=1, ステップ⑧)。

【0159】すると、シリアル通信装置11は、スタートビットの検出を待って(ステップ⑨, 「YES」)バイトデータの受信を行なう(ステップ“10”)。この場合、受信バッファ23がフルになった場合は、CPU1

2に対して割り込みRBFUL_INTを発生させて、受信データの引上げを要請する(ステップ“11”)。

【0160】データの受信中にフレーミングエラー(ストップビット検出無し)が発生すると(ステップ“12”, 「YES」)、CPU12に対して割り込みRFERR_INTを発生させてステップ“17”でエラー処理を行なわせる。また、受信中にステップ“16”において受信フレームタイムアウトが発生すると(「YES」)、CPU12に対してタイムアウトエラー割り込みRX_FTOERR_INTを発生させる。

【0161】そして、データの受信が完了すると(ステップ“13”, 「YES」)、CPU12は受信を禁止し(RE=0)、受信データ中に含まれているFCCをチェックする(ステップ“14”)。FCCの計算結果がOKであれば(「YES」)処理を終了し、結果がNGであればステップ“17”に移行する。

【0162】<受信処理>図46は受信処理のフローチャート、図47は同通信シーケンスを示す。ステップ①で行なう各種設定は、基本的に送信処理の場合と同様である(但し、DMAC14は受信に使用する)。そして、シリアル通信装置11は、汎用入力UIN3においてHBREQがハイレベルになったことを検出すると(ステップ②, 「YES」)、CPU12に対して割り込みINP3_IRを発生させる。すると、CPU12は、送信データサイズを任意に設定して送信を許可する(TE=1, ステップ③)。

【0163】シリアル通信装置11は、バイトデータの送信(データあり?の問い合わせ)を開始するが(ステップ④)、送信バッファ32が空になった場合は、CPU12に対して割り込みTBEMP_INTを発生させて、CPU12に送信データをセットさせる(ステップ⑤)。送信が終了すると(ステップ⑤, 「YES」)終了割り込みにより、CPU12は、受信設定(ステップ⑦, 送信禁止、割り込みクリア、DMA使用、受信データサイズ設定)を行ない、受信を許可する(ステップ⑧)。

【0164】以降の受信処理ステップ⑨～“13”は、送信処理における受信処理ステップ⑨～“14”と略同様に行なわれるが、受信バッファ23がフルになった場合は、DMAC14に対して転送要求DREQを発生させて、受信データの転送を要請する。

【0165】CPU12は、ステップ“13”において、FCCの計算結果がOKであれば(「YES」)、受信データがSDHであるか否かを判断し(ステップ“14”)、SDHである場合は(「YES」)その後続く受信データフレームを受信するためステップ⑦に戻る。SDHでない場合は(「NO」)データフレームか否かを判断する(ステップ“15”)。

【0166】即ち、SDHに続くデータフレームを受信した場合、CPU12はステップ“15”で「YES」と判断して、応答ACKを送信するため送信を開始する

(ステップ“16”)。そして、シリアル通信装置11がステップ“17”, “18”においてステップ④, ⑤と同様にして応答ACKを送信すると受信処理を終了する。また、ステップ“19”, “20”におけるフレーミングエラー処理は、送信処理におけるステップ“15”, “16”と同様に行われる。

【0167】以上のように第6実施例によれば、シリアル通信装置11を、HIT64のモジュールとして機能させるようにしたので、通信にクロックを使用しないようなプロトコルにも対応させることができる。

【0168】(第7実施例) 図48乃至図52は本発明の第7実施例である。第7実施例は、シリアル通信装置11をCHIP (Centralized High-level Information control Protocol) に適用した場合であり、図48は、シリアル通信装置11をCHIPモジュールとして機能させる場合における入出力信号の割当状態を示すものである。

MDT : 送信データ (TX)

SMD : 受信データ (RX)

BREQ : 受信要求信号 (UIN3)

【0169】<送信処理> 図49は送信処理のフローチャート、図50は同タイミングチャートを示す。先ず、ステップ①では、CPU12は以下の設定を行なう。

(送信初期化)

- ・非同期プロトコル、全二重通信。TXの初期値=1。
- ・スタート/ストップビット、夫々1ビット。

(汎用入力設定)

- ・汎用入力1, 2 (UIN1, 2) をS/W設定。ハイレベル検出。夫々“1”をセット。
- ・汎用入力3を外部入力設定。ハイレベル検出。

【0170】(送信制御)

- ・TX_FSTA にTEを選択。
- ・TX_BSTA に汎用入力2 (UIN2) を選択。
- ・TX_FEND にデータサイズカウンタを選択。データサイズ任意 (但し、FH~DCCまでのデータサイズ)。
- ・バイトサイズカウンタをTX_FSTA でセット。バイトサイズN=8。

(送信マルチカウンタ設定)

- ・TCKカウント, TX_FEND でカウンタセット, カウントサイズ=15。→DMYコード出力に用いる。

【0171】(受信制御)

- ・RX_FSTA に汎用入力1 (UIN1) を選択。
- ・RX_BSTA にパターンマッチ1を選択。
- ・RX_FEND に受信マルチカウンタを選択。
- ・バイトサイズカウンタ, RX_BSTA でセット, バイトデータのサイズN=8。

(パターンマッチ設定)

- ・第7実施例と同様にスタートビット検出。

(エラー)

- ・送信汎用マッチにマルチカウンタ出力を選択。割り込

み要因に設定。

- ・バイトタイムアウトを、TCK15クロック相当のカウント値に設定。

- ・FCC設定。

(DMA初期設定)

以上までが、ステップ①に相当する。

【0172】次に、CPU12は、ステップ②においてDMAC14のカウンタ設定を行なうと共に送信を許可する (TE=1)。ステップ③, ④は、シリアル通信装置11によりバイトデータの送信 (FH, DATA, FCC) が行われる。FCCまでのデータ送信が終了すると (ステップ④, 「YES」) トリガ信号TX_FEND が発生し、送信マルチカウンタがカウントを開始する。

【0173】そして、送信マルチカウンタがTCKを15カウントすることでカウンタマッチ信号が出力されると (ステップ⑤, 「YES」) DMYコードが送信されたことになり、シリアル通信装置11は、CPU12に対する割り込みTMC_INT を発生させる。

【0174】CPU12は、前記割り込みの発生を受けて受信設定 (送信禁止, 割り込みクリア) 及び受信開始 (受信許可RE=1) を行う (ステップ⑥, ⑦)。すると、シリアル通信装置11は受信を開始して、パターンマッチ1によりスタートビットを検出すると (ステップ⑧, 「YES」: RX_BSTA 発生) スレーブ側より送信されるバイトデータの受信を開始する (ステップ⑨, “1”)。

【0175】ステップ⑨において受信バッファ23がフルの場合は割り込みによってCPU12にデータの引上げを要請し (ステップ“10”)、フレーミングエラーが発生した場合は (ステップ“11”, 「YES」) CPU12にエラー処理を実行させる (ステップ“12”)。そして、ステップ“13”においてバイトデータの受信終了後、TCK15クロック分の無通信状態が生じると受信バイトタイムアウトが発生し (RX_FEND 発生)、CPU12に対して割り込みRX_BTOERR_INT を発生させて受信を終了させる。

【0176】<受信処理> 図51は受信処理のフローチャート、図52は同タイミングチャートを示す。ステップ①で行なう各種設定は、基本的に送信処理の場合と同様である (但し、DMAC14は受信に使用する。また、汎用入力UIN3を、外部入力でロウレベル検出 (BREQ) に設定。)。そして、シリアル通信装置11は、汎用入力UIN3においてBREQがロウレベルになったことを検出すると、CPU12に対して割り込みINP3_IR を発生させる (ステップ②)。すると、CPU12は送信を許可 (TE=1) する (ステップ③)。

【0177】シリアル通信装置11は、ステップ④, ⑥において送信処理におけるステップ③~⑤と同様にバイトデータ (但し、FH, FCC, DMY) の送信を行なうが、送信バッファ32が空になった場合は、CPU1

2 に対して割り込み TBEMP_INT を発生させて、CPU 12 に送信データをセットさせる (ステップ⑤)。

【0178】また、ステップ⑧～“14”における受信処理も、基本的に送信処理におけるステップ⑥～“13”と同様に行われる。但し、受信バッファ 23 がフルになった場合は、DMAC 14 に対して転送要求 DREQ を発生させて、受信データの転送を要請する。以上のように第 7 実施例によれば、シリアル通信装置 11 を、CHIP のプロトコルにも対応させることができる。

【0179】(第 8 実施例) 図 53 乃至図 57 は本発明の第 8 実施例である。第 8 実施例は、シリアル通信装置 11 を CCB (Computer Control Bus) に適用した場合であり、図 43 は、シリアル通信装置 11 を CCB モジュールとして機能させる場合における入出力信号の割当状態を示すものである。

CL : 同期クロック (CLKOUT)
DI : マスタ送信データ (TX)
DO : マスタ受信データ (RX)
CE : データ制御信号 (UOUT1)
INT : CCB デバイスからの割り込み信号 (UIN3)

【0180】<送信処理> 図 54 は送信処理のフローチャート、図 55 は同タイミングチャートを示す。先ず、ステップ①では、CPU 12 は以下の設定を行なう。

(送信初期化)

- ・同期プロトコル、全二重通信。TX の初期値 = 1。
- ・CLKOUT よりシフトクロック SCK を出力。

(汎用入力)

- ・汎用入力 2 (UIN2) を S/W 設定。ハイレベル検出。“1”をセット。

(汎用出力)

- ・汎用出力 1 (UOUT1) のセット、リセットを S/W 設定。

【0181】(送信制御)

- ・TX_FSTA に (UIN2 & not TBEMPT) を選択。
- ・TX_BSTA に汎用入力 2 (UIN2) を選択。
- ・TX_FEND にデータサイズカウンタを選択。
- ・バイトサイズカウンタ、TX_FSTA でセット、バイトデータのサイズ N = 8。

(DMA 初期設定)

(送信許可)

- ・データサイズ 8 ビット、送信許可 (TE = 1)。

以上までが、ステップ①に相当する。

【0182】次に、CPU 12 は、ステップ②において送信バッファ 32 に送信先デバイスのアドレス及び R/W を書込む。すると、条件 (UIN2 & not TBEMPT) が成立し、TX_FSTA 及び TX_BSTA が発生し、シリアル通信装置 11 はアドレス及び R/W を送信する (ステップ③)。

【0183】8 ビットデータを送信すると TX_BEND 及び

TX_FEND が発生し、シリアル通信装置 11 は CPU 12 に対して送信終了割り込みを発生させる。すると、CPU 12 は、データ送信用の設定を行なう (ステップ④)。即ち、一旦送信を禁止 (TE = 0) してから送信データサイズを設定し、データ制御信号 CE を出力 (セット) するためレジスタに書込みを行なう。すると、シリアル通信装置 11 は、UOUT1 をロウレベルからハイレベルに遷移させてデータ制御信号 CE を出力する (ステップ⑤)。

【0184】続いて、CPU 12 は DMAC 14 のカウンタをセットして送信を許可する (TE = 1, ステップ⑥)。すると、シリアル通信装置 11 によりバイトデータの送信が行われる (ステップ⑦, ⑧)。バイトデータ送信が終了すると (ステップ⑧, 「YES」) CPU 12 に対して送信終了割り込みを発生させ、CPU 12 は、送信を禁止 (TE = 0) してから、データ制御信号 CE の出力を停止 (リセット) するためレジスタに書込みを行なう (ステップ⑨)。すると、シリアル通信装置 11 は、UOUT1 をハイレベルからロウレベルに遷移させてデータ制御信号 CE の出力を停止する (ステップ “10”)。

【0185】<受信処理> 図 56 は受信処理のフローチャート、図 57 は同タイミングチャートを示す。ステップ①では、CPU 12 は以下の設定を行なう。

(送信初期化) : 基本的に送信処理と同様。

(汎用入力)

- ・汎用入力 1, 2 (UIN1, 2) を S/W 設定。ハイレベル検出。夫々 “1” をセット。
- ・汎用入力 3 (UIN3) を外部入力設定。ロウレベル検出。

(汎用出力) : 送信処理と同様。

(受信制御)

- ・RX_FSTA に RE を選択。
- ・RX_BSTA に汎用入力 1 (UIN1) を選択。
- ・RX_FEND にデータサイズカウンタを選択。受信データサイズは任意。
- ・バイトサイズカウンタ、RX_BSTA でセット、バイトデータのサイズ N = 8。

【0186】(送信制御)

- ・TX_FSTA に TE を選択。
- ・TX_BSTA に汎用入力 2 (UIN2) を選択。
- ・TX_FEND にデータサイズカウンタを選択。
- ・バイトサイズカウンタ、TX_FSTA でセット、バイトデータのサイズ N = 8。

(DMA 初期設定)

以上までが、ステップ①に相当する。

【0187】次に、シリアル通信装置 11 は、UIN3 において接続先デバイスからの割り込み信号 INT を検出するまで待機し (ステップ②)、割り込み信号 INT を検出すると (「YES」) CPU 12 に対して割り込み

INP3_INTを発生させる。以降のステップ③～“11”は、
＜送信処理＞におけるステップ②～“10”と基本的に同
様である。但し、ステップ⑧において受信するのは、接
続先デバイスが送信したデータである。

【0188】以上のように第8実施例によれば、シリアル
通信装置11を、CCBプロトコルにも対応させること
ができる。そして、シリアル通信装置11の、送信側
のトリガ信号発生手段たる出力回路63、66、67の
トリガ信号発生条件として、通信制御信号の遷移状態を
カウントするカウンタより出力されるカウンタマッチ信号、
送信データのサイズをカウントするカウンタより出力
されるカウンタマッチ信号、外部より付与される外部
トリガ信号を、適宜組み合わせる選択するように構成し
た。

【0189】また、受信側のトリガ信号発生手段たる出
力回路57、60、61におけるトリガ信号発生条件と
して、通信制御信号の遷移状態をカウントするカウンタ
より出力されるカウンタマッチ信号、受信データのサイ
ズをカウントするカウンタより出力されるカウンタマッ
チ信号、受信データのパターンが特定データパターンに
一致した場合にコンパレータより出力されるパターンマ
ッチ信号、外部より付与される外部トリガ信号の各条件
を適宜組み合わせる選択するように構成した。

【0190】従って、シリアル通信の様々なプロトコル
に対応して、適切なタイミングでトリガ信号を発生させ
ることができる。そして、外部トリガ信号としては、C
PU12がシリアル通信装置11の内部レジスタに書き込
みを行うことで発生させるトリガ信号と、外部より汎用
入力UIN1～3に与えられる通信制御信号の遷移状態
に基づいて発生されるトリガ信号とを選択可能に構成し
たので、各プロトコルに応じて、CPU12のプログラ
ム動作や、汎用入力UIN1～3に与えられる通信制御
信号の遷移状態などのタイミングでトリガ信号を発生さ
せるように選択することができ、より柔軟にプロトコル
対応を行うことが可能となる。

【0191】本発明は上記し且つ図面に記載した実施例
にのみ限定されるものではなく、次のような変形または
拡張が可能である。接続先デバイスは、カーオーディオ
装置102に限ることなく、シリアル通信を行うもので
あれば何でもよい。以上の第1～第8実施例では、トリ
ガ信号の発生条件について選択しなかった要因がある
が、第1～第8実施例において示したプロトコルの範囲
で対応することを考えればそれらは必ずしも必要ではな
い。また、第1～第8実施例において示したプロトコル
について全て対応する必要がない場合には、不要な要因
は適宜削除すれば良い。更に、第1～第8実施例では示
さなかったその他のプロトコルについて、選択しなかつ
た要因を用いることで対応することが想定できる。

【0192】受信ステートマシンにおける“エラー”ス
テートは、エラー処理が必要なプロトコルについてのみ

設ければ良い。また、フラグデータやスタート/ストッ
プビットの挿入、通信データ中のビット挿入及び削除、
パリティの付加などの機能も、対応するプロトコル範囲
に応じて選択的に設ければ良い。例えばスタートビット
だけを挿入するようにしても良い。データ転送装置は必
要に応じて設ければ良く、データ転送装置を設けない場
合は、送信バッファエンプティ、受信バッファフルが発
生する毎にホストに対して割込みを発生させ、送信デー
タの書き込み、受信データの引上げをホストに要請すれば
良い。ホストは、CPU12に限ることなく、その他D
SPなどの汎用演算処理装置であれば良い。

【図面の簡単な説明】

【図1】本発明の第1実施例であり、マルチプロトコル
型シリアル通信装置の全体構成を概略的に示す機能ブロ
ック図

【図2】シリアル通信装置で取り扱われるデータ列及び
トリガ信号の定義を示すチャート

【図3】受信時における各種クロック信号の定義を示す
タイミングチャート

【図4】受信バッファを中心とする詳細な構成を示す図

【図5】(a)は受信側のパターンマッチ回路の構成を
示す図、(b)はパターンマッチ回路に対してイネーブル
信号を出力する構成を示す図

【図6】カウンタマッチ検出回路の構成を示す図

【図7】受信ステートマシンの状態遷移図

【図8】受信側のフレームスタート信号の出力回路の構
成を示す図

【図9】受信側のフレームエンド信号の出力回路の構成
を示す図

【図10】受信側のバイトスタート信号の出力回路の構
成を示す図

【図11】送信バッファを中心とする詳細な構成を示す
図

【図12】送信側のフレームスタート信号の出力回路の
構成を示す図

【図13】送信側のフレームエンド信号の出力回路の構
成を示す図

【図14】送信側のバイトスタート信号の出力回路の構
成を示す図

【図15】汎用的に入力される信号を内部制御に使用す
るための構成を概略的に示す図

【図16】内部で汎用化したトリガ信号等に基づいて、
シリアル通信装置より外部に出力される信号を各プロト
コルに応じて生成するための構成を概略的に示す図

【図17】汎用出力信号の出力例を示すタイミングチャ
ート

【図18】シリアル通信装置を、TAB2プロトコルに
適用した場合のシステム構成を示す図

【図19】CPUがカーオーディオ装置に対してデータ
を送信する場合のフローチャート

【図 20】TAB 2 の送信タイミングチャート

【図 21】CPU がカーオーディオ装置よりデータを受信する場合のフローチャート

【図 22】TAB 2 の受信タイミングチャート

【図 23】本発明の第 2 実施例であり、シリアル通信装置を RS-232C モジュールとして機能させる場合における入出力信号の割当状態を示す図

【図 24】CPU がデータを送信する場合のフローチャート

【図 25】送信タイミングチャート

【図 26】送信ステートマシンのより詳細な状態遷移図

【図 27】CPU がデータを受信する場合のフローチャート

【図 28】受信タイミングチャート

【図 29】本発明の第 3 実施例であり、シリアル通信装置を HDLC モジュールとして機能させる場合における入出力信号の割当状態を示す図

【図 30】図 24 相当図

【図 31】図 25 相当図

【図 32】図 26 相当図

【図 33】図 27 相当図

【図 34】本発明の第 4 実施例であり、シリアル通信装置を IIC モジュールとして機能させる場合における入出力信号の割当状態を示す図

【図 35】図 24 相当図

【図 36】図 25 相当図

【図 37】図 26 相当図

【図 38】図 27 相当図

【図 39】本発明の第 5 実施例であり、シリアル通信装置を IIS モジュールとして機能させる場合における入出力信号の割当状態を示す図、(a) はクロックマスタの場合、(b) はクロックスレーブの場合

【図 40】図 24 相当図

【図 41】図 25 相当図

【図 42】図 26 相当図

【図 43】本発明の第 6 実施例であり、シリアル通信装置を HIT 64 モジュールとして機能させる場合におけ*

* 入出力信号の割当状態を示す図

【図 44】図 24 相当図

【図 45】送信処理のシーケンス図

【図 46】図 26 相当図

【図 47】受信処理のシーケンス図

【図 48】本発明の第 7 実施例であり、シリアル通信装置を CHIP モジュールとして機能させる場合における入出力信号の割当状態を示す図

【図 49】図 24 相当図

10 【図 50】図 25 相当図

【図 51】図 26 相当図

【図 52】図 27 相当図

【図 53】本発明の第 8 実施例であり、シリアル通信装置を CCB モジュールとして機能させる場合における入出力信号の割当状態を示す図

【図 54】図 24 相当図

【図 55】図 25 相当図

【図 56】図 26 相当

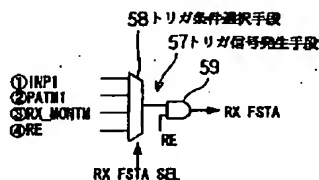
【図 57】図 27 相当図

20 【図 58】従来技術を示す図 18 相当図

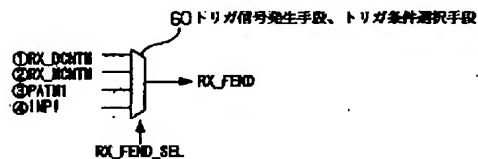
【符号の説明】

11 はマルチプロトコル型シリアル通信装置、12 は CPU (ホスト)、14 は DMA コントローラ (データ転送装置)、23 は受信バッファ (内部バッファ)、30 は受信ステートマシン、32 は送信バッファ (内部バッファ)、38 は受信ステートマシン、57 は出力回路 (トリガ信号発生手段)、58 はマルチプレкса (トリガ条件選択手段)、60、61 は出力回路 (トリガ信号発生手段、受信トリガ条件選択手段)、62 はビット挿入部 (ビット挿入手段)、63 は出力回路 (トリガ信号発生手段)、64 はマルチプレкса (トリガ条件選択手段)、66 は出力回路 (トリガ信号発生手段、トリガ条件選択手段)、67 は出力回路 (トリガ信号発生手段)、68 はマルチプレкса (トリガ条件選択手段)、100 はメモリ、101 は ECU (マイクロコンピュータ)、102 はカーオーディオ装置 (接続先デバイス) を示す。

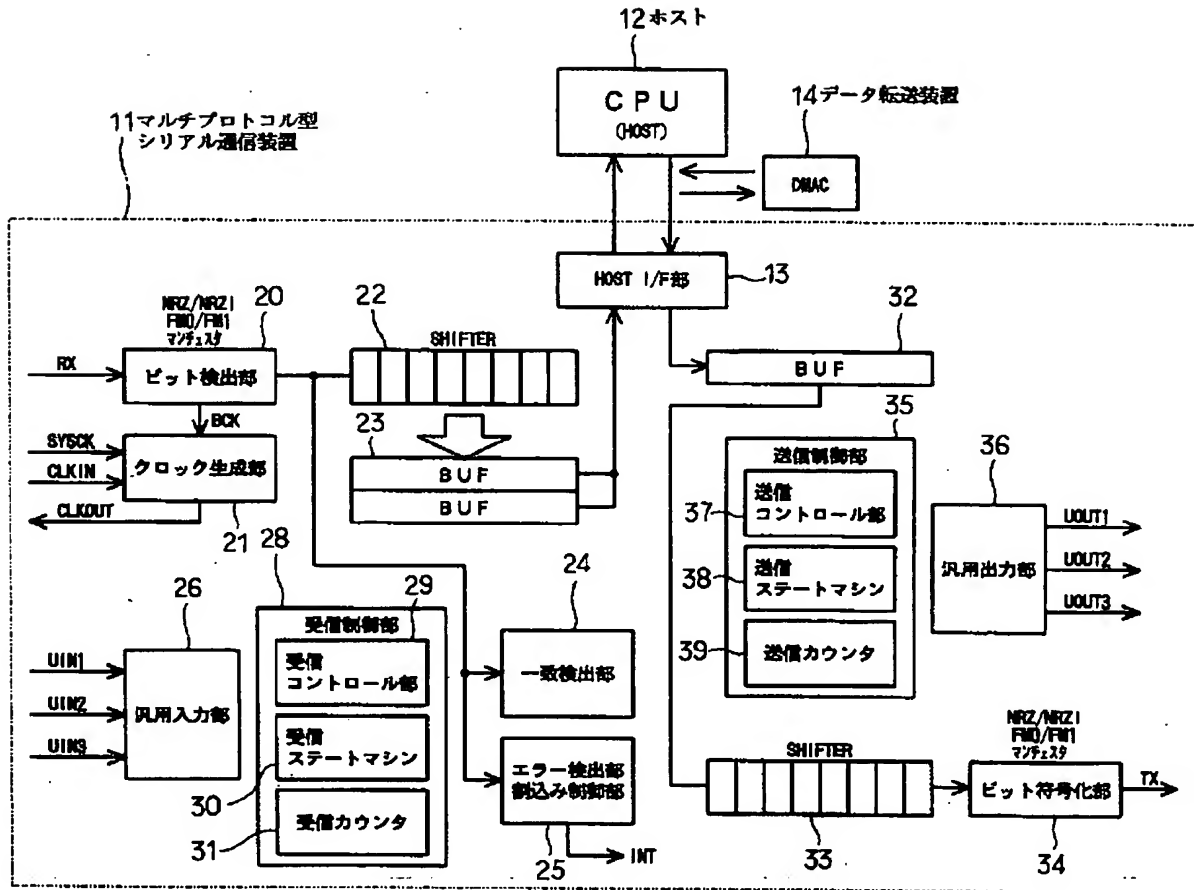
【図 8】



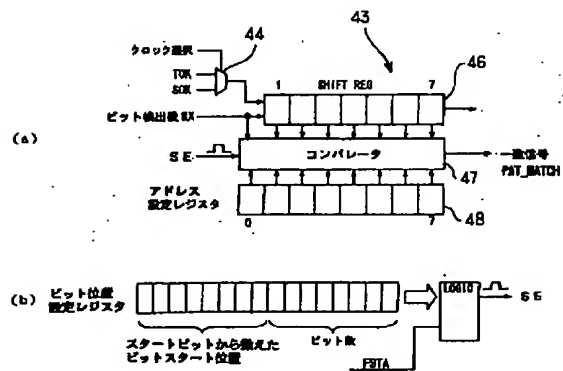
【図 9】



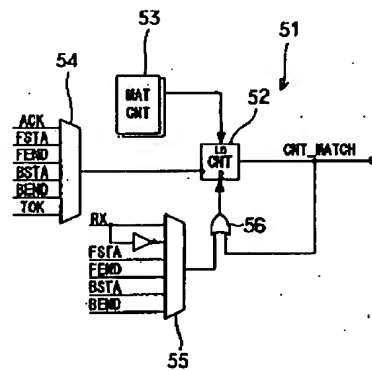
【図1】



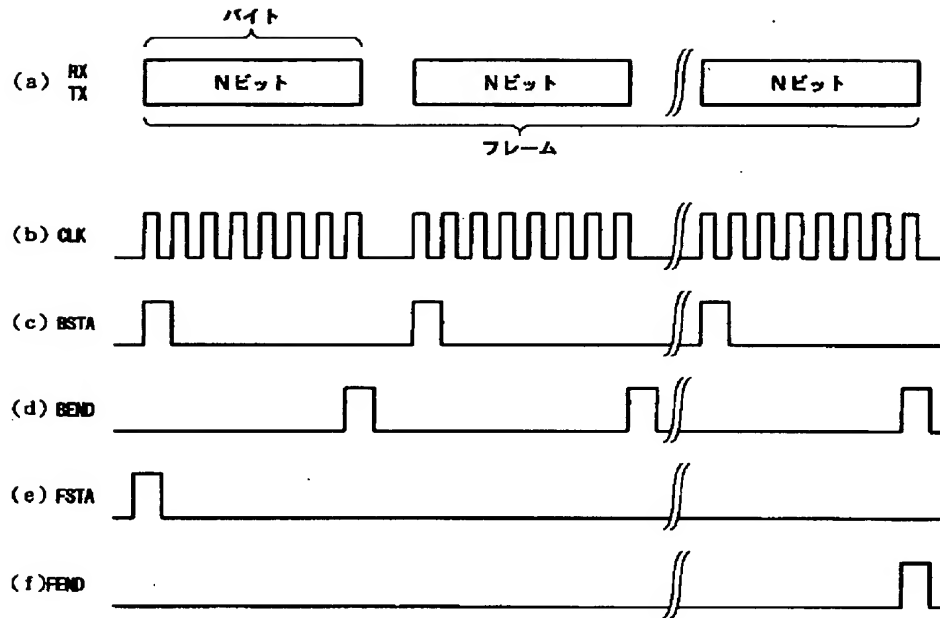
【図5】



【図6】

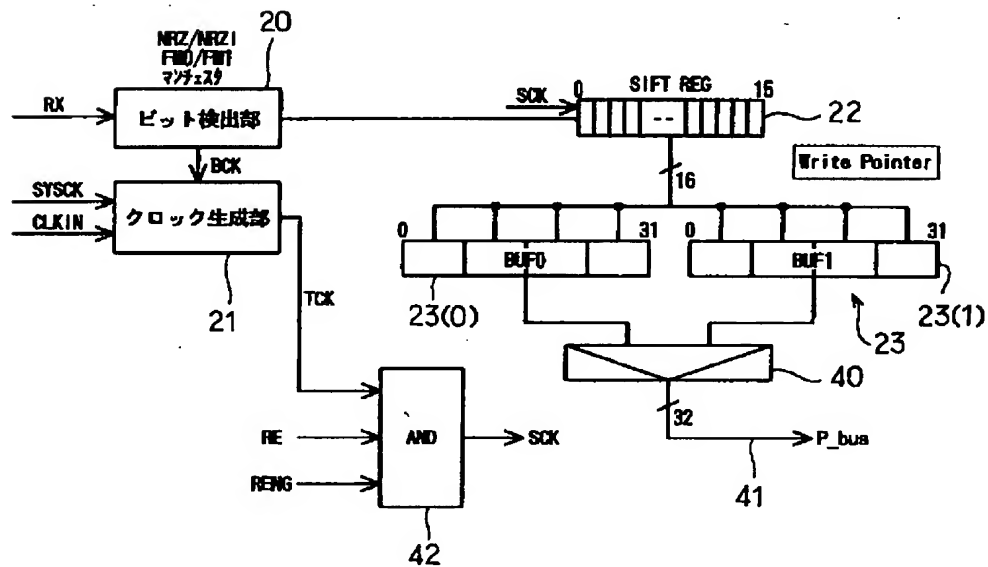


【図2】

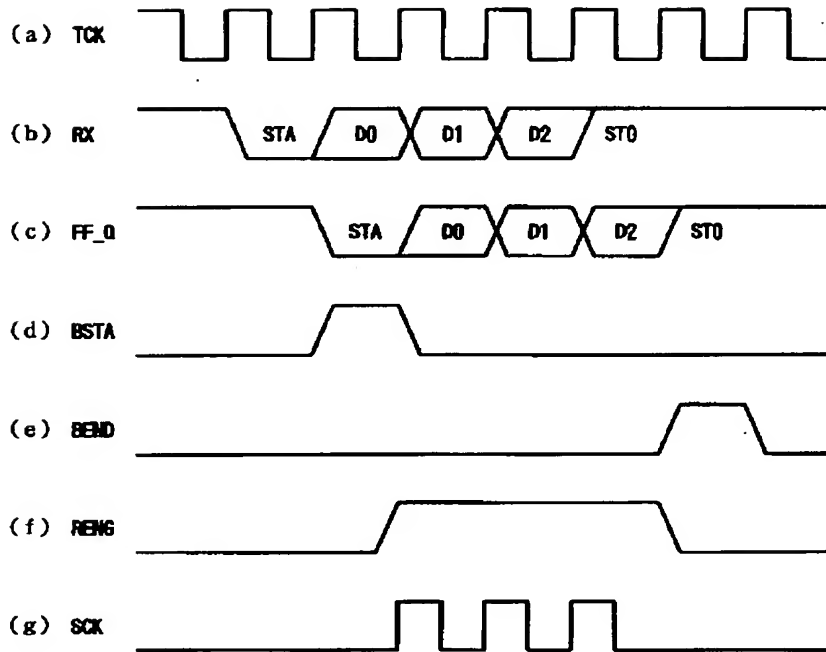


RX : 受信データ
 BSTA : バイトスタート
 FSTA : フレームスタート
 CLK : クロック
 BEND : バイトエンド
 FEND : フレームエンド

【図4】



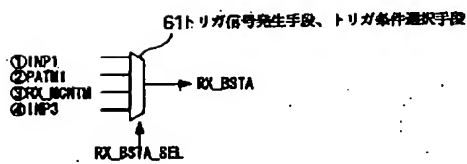
【図3】



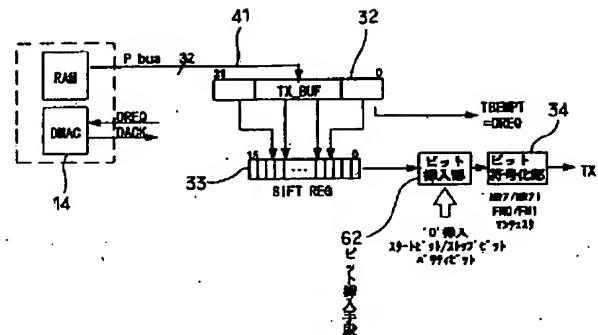
TCK : 伝送クロック
 FF_Q : RXのTCK同期信号
 BSTA : バイトスタート信号
 RENG : 受信中信号

RX : 受信データ
 BEND : バイトエンド信号
 SCK : データシフトクロック

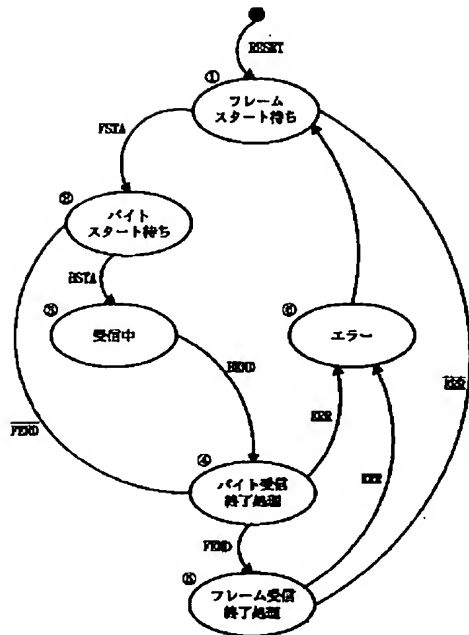
【図10】



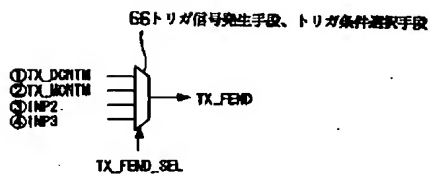
【図11】



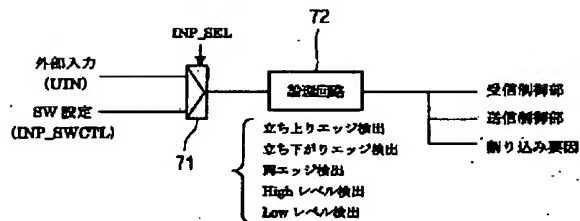
【図7】



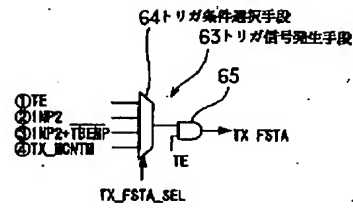
【図13】



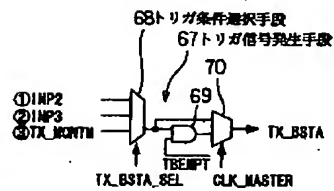
【図15】



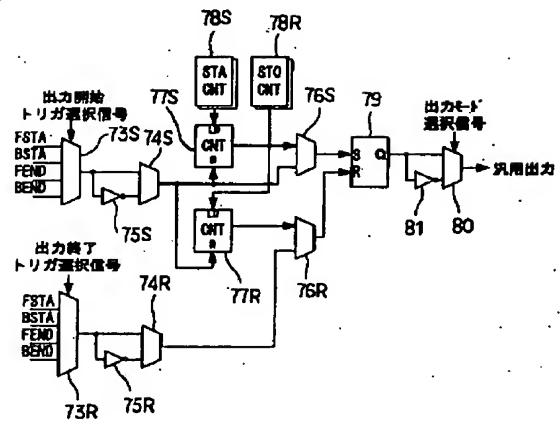
【図12】



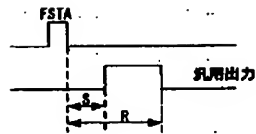
【図14】



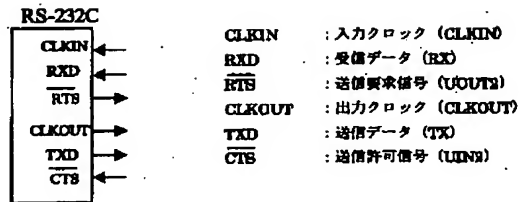
【図16】



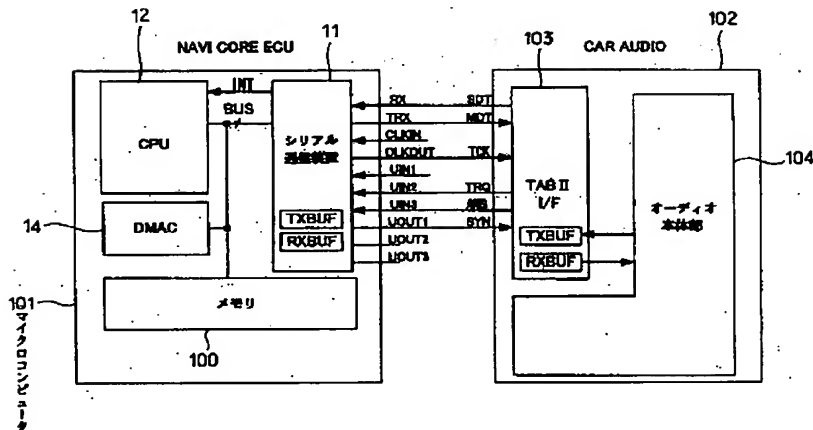
【図17】



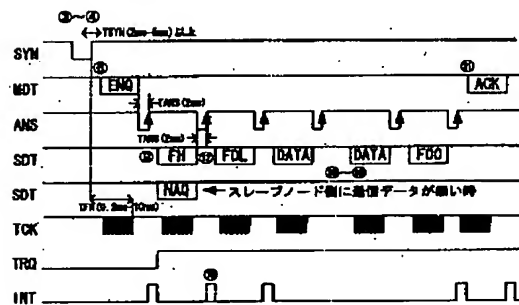
【図23】



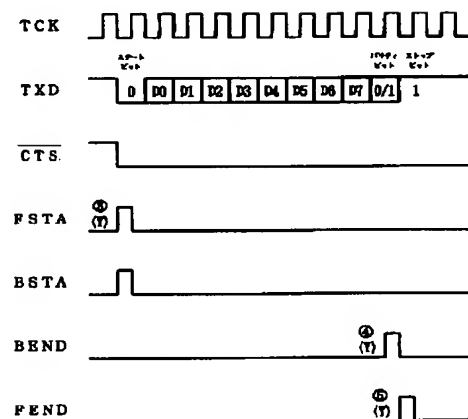
【図18】



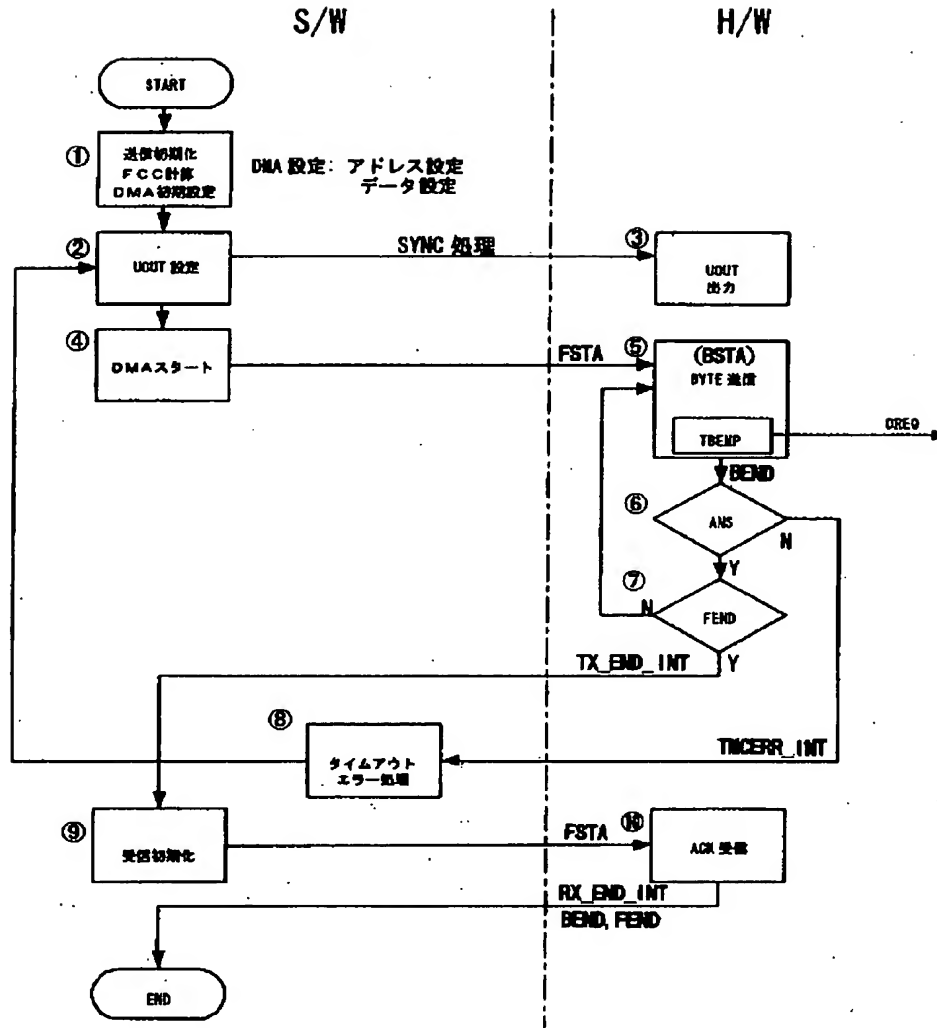
【図22】



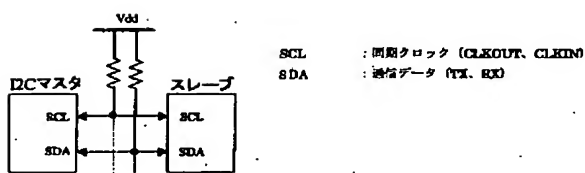
【図25】



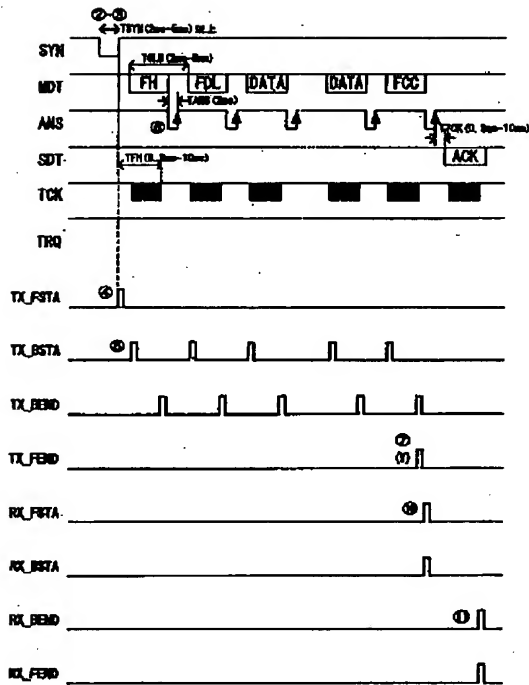
【図19】



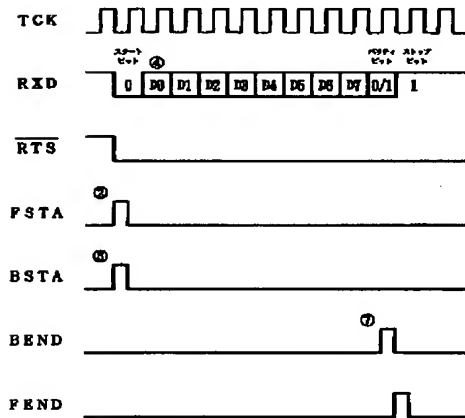
【図34】



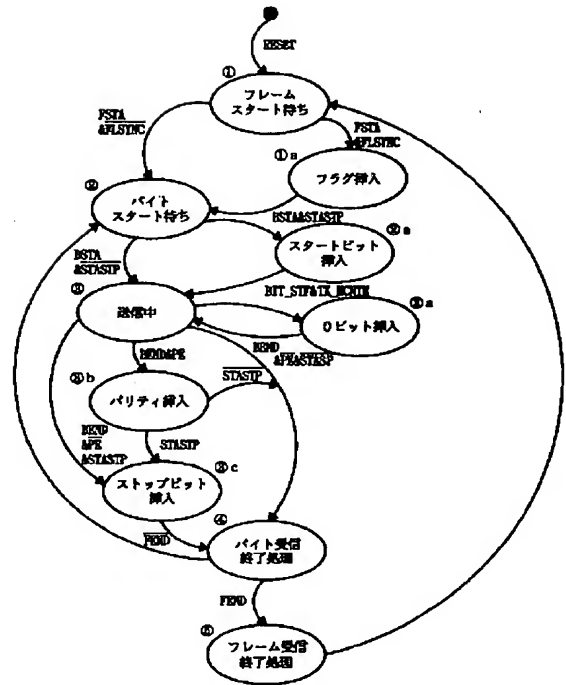
【図20】



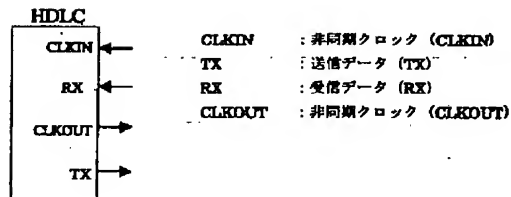
【図28】



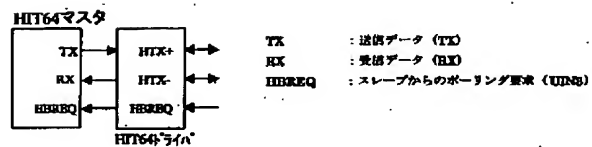
【図26】



【図29】

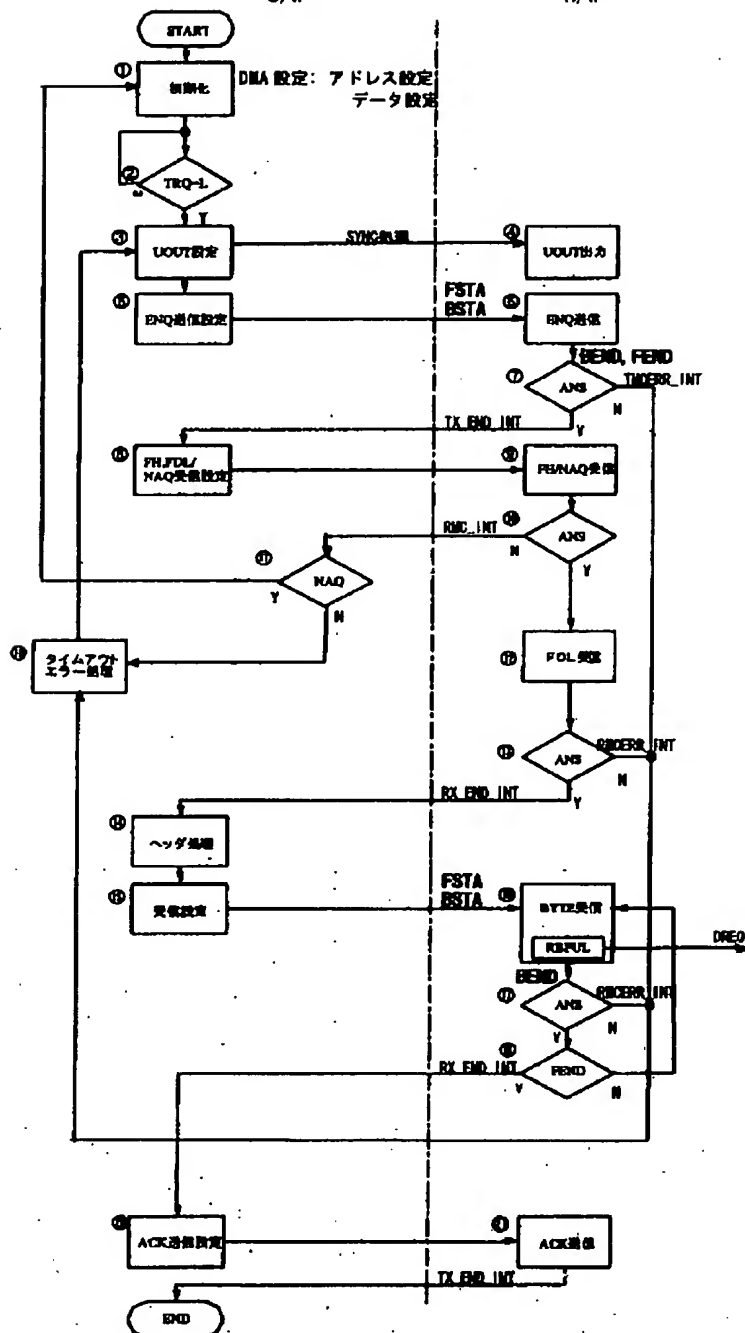


【図43】

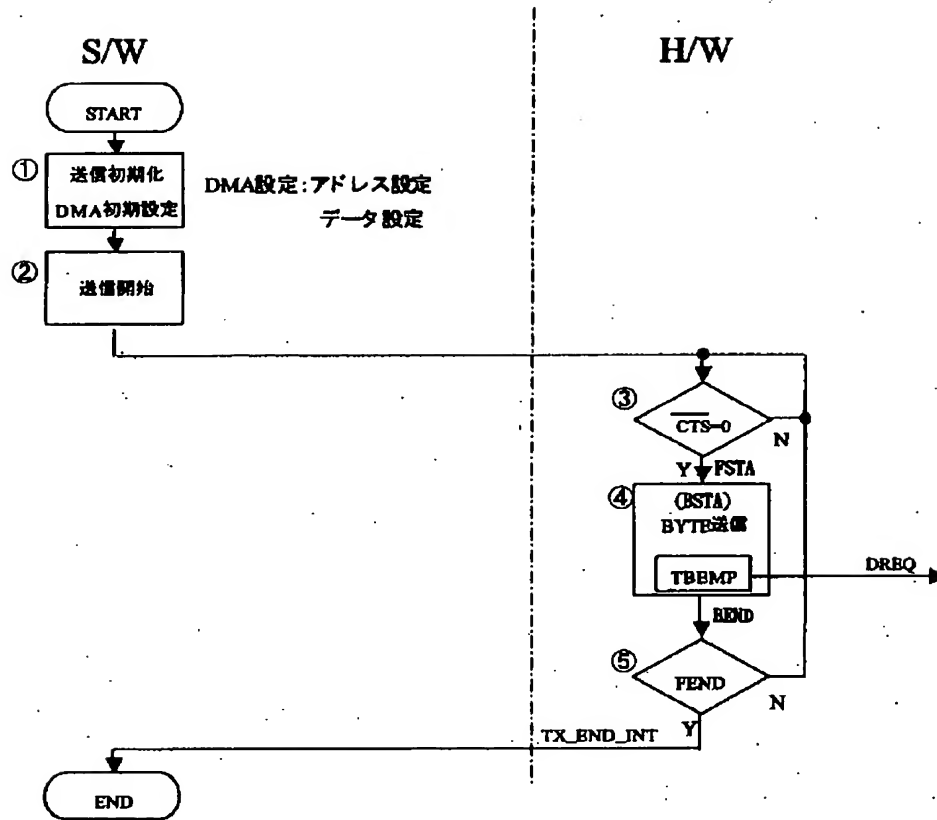


S/W

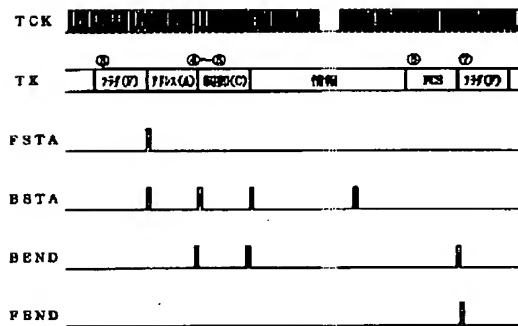
H/W.



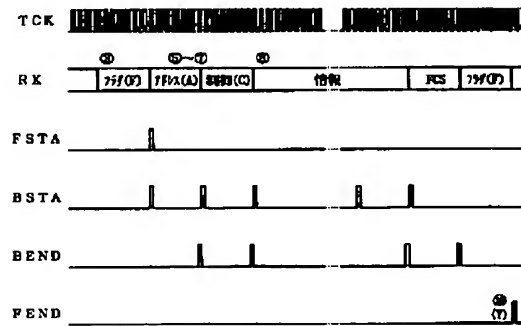
【図24】



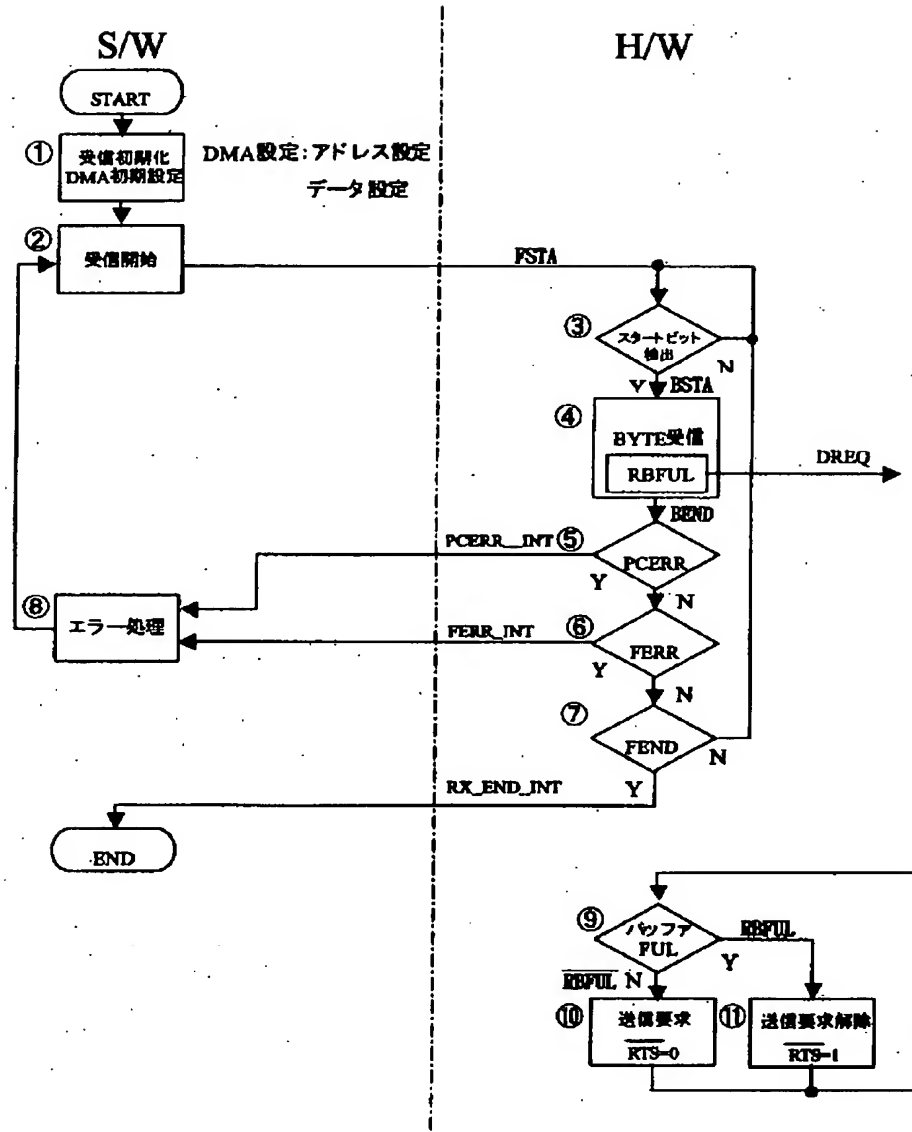
【図31】



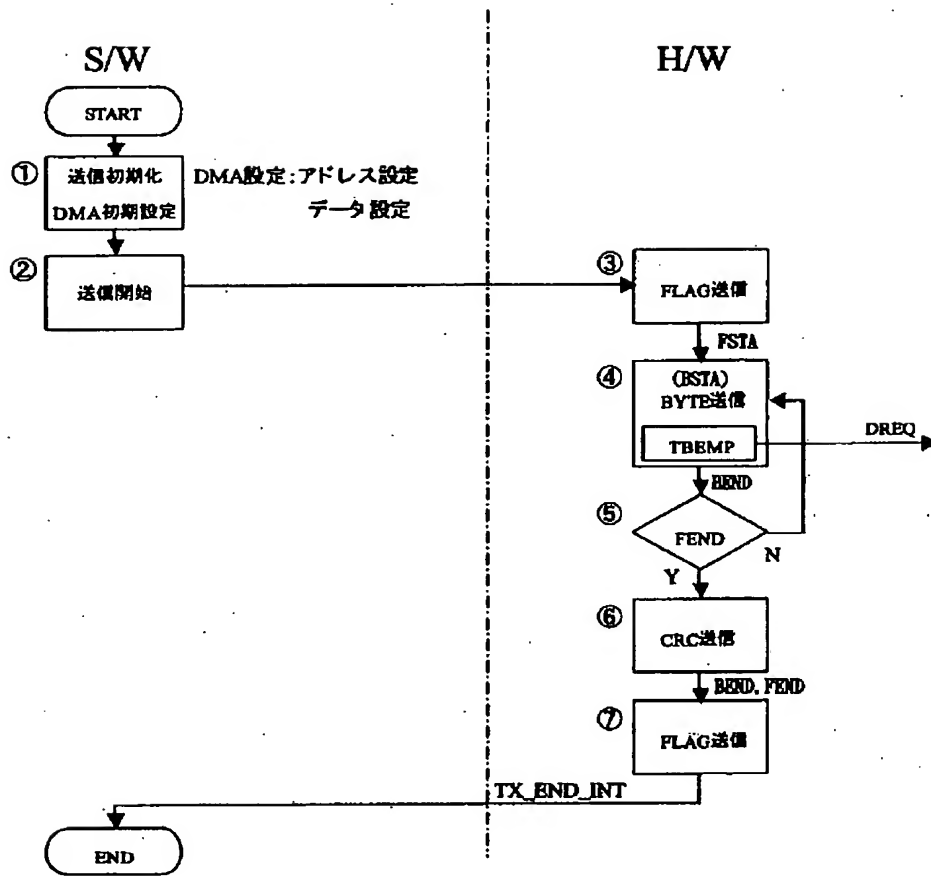
【図33】



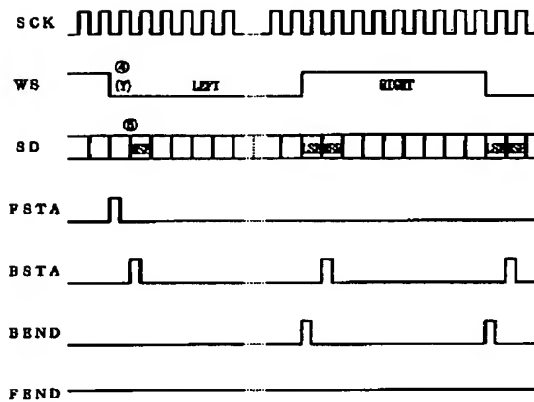
【図27】



【図30】



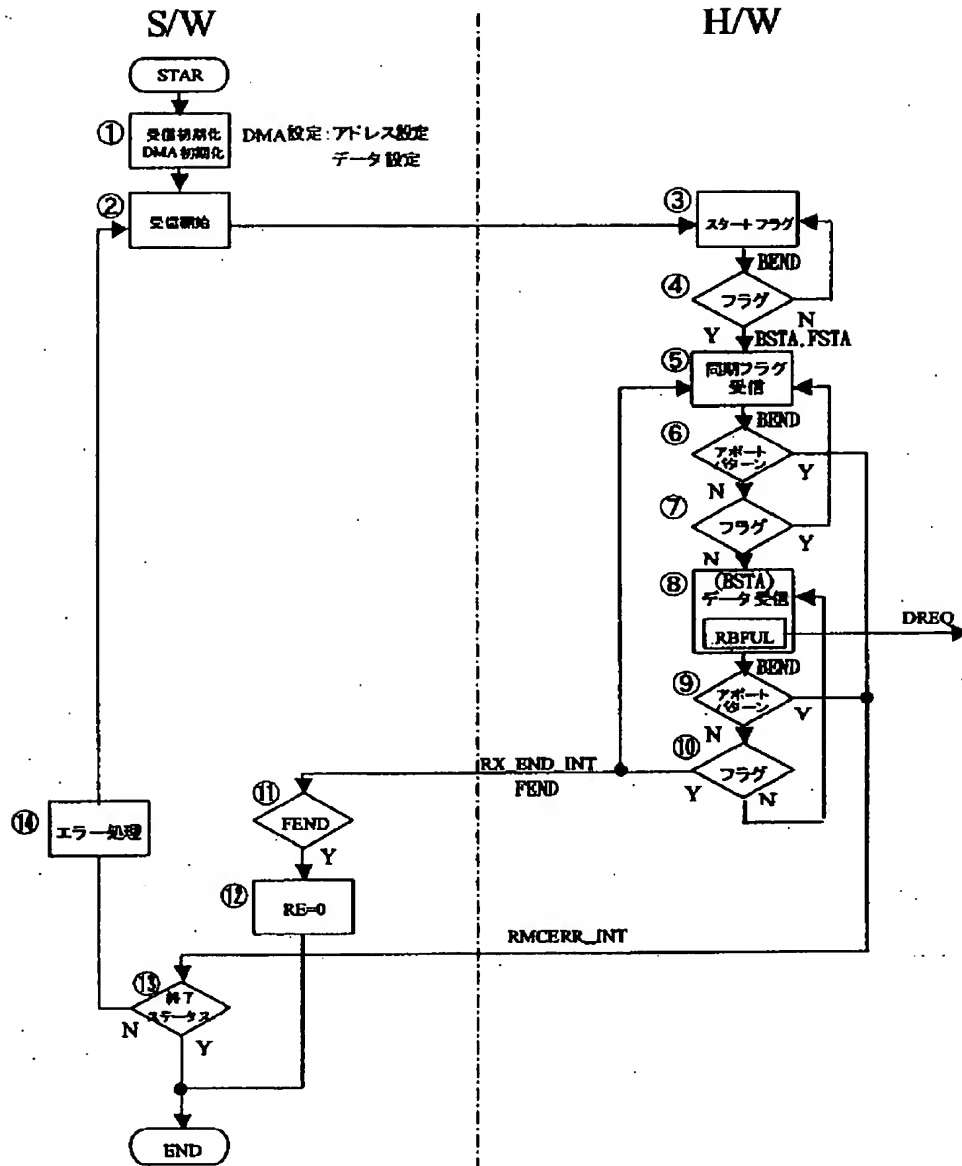
【図41】



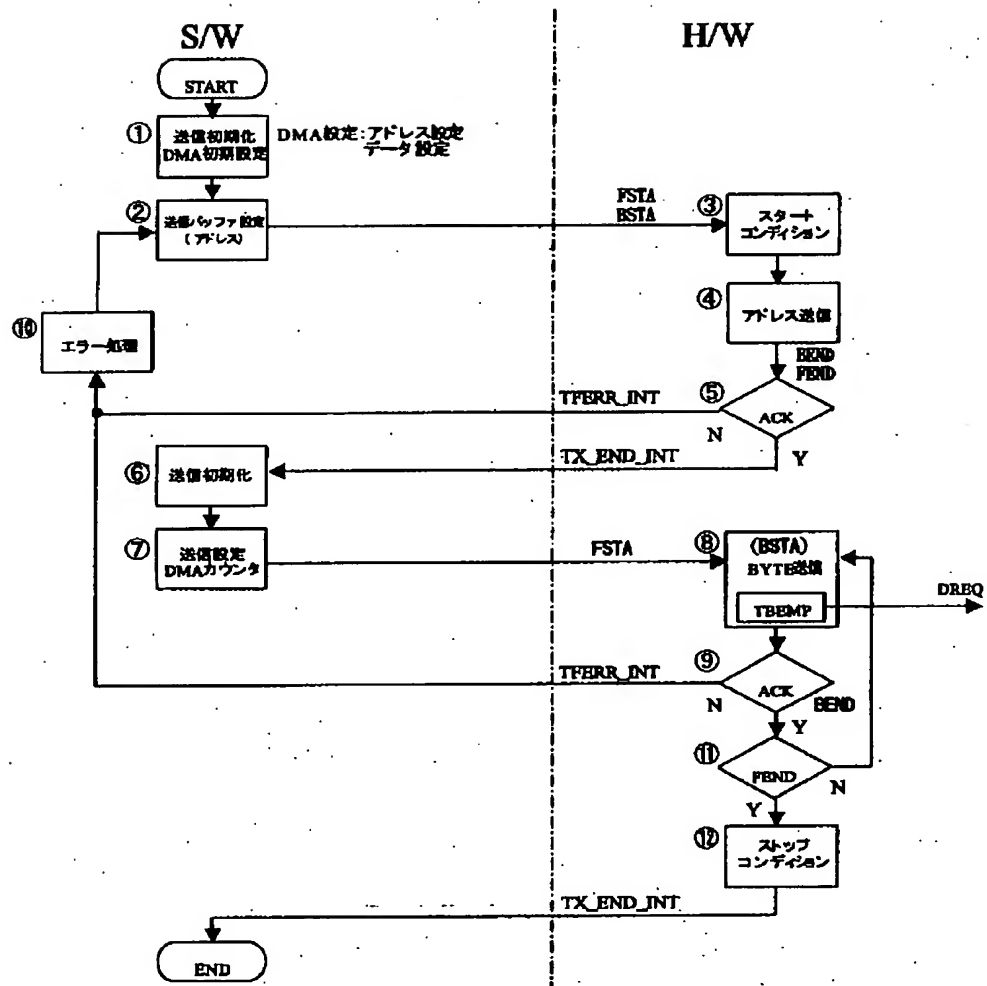
【図48】



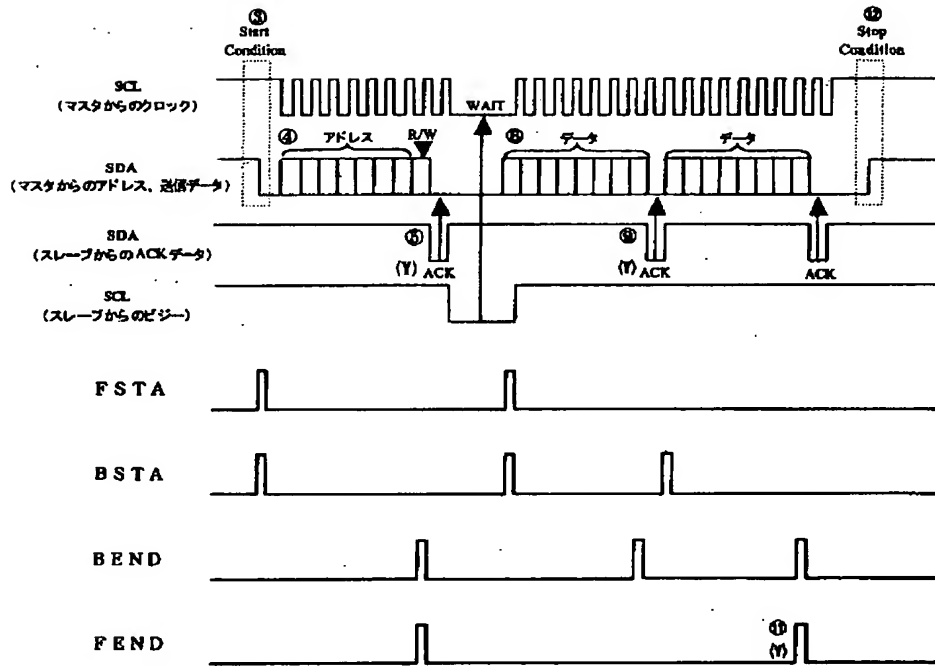
【図32】



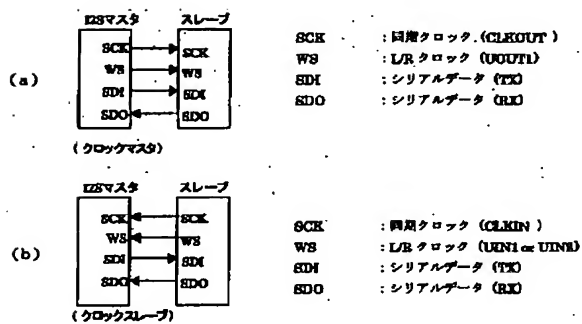
【図35】



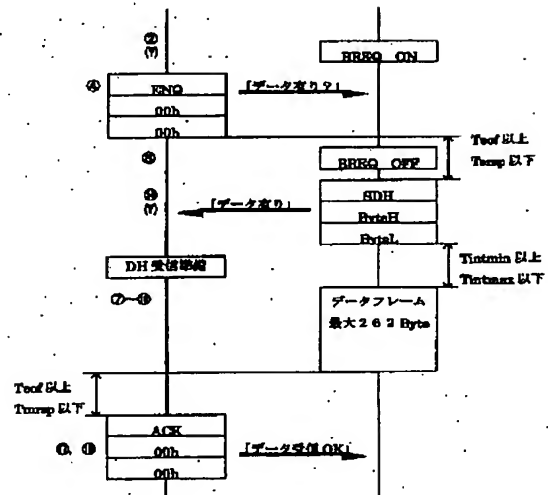
【図36】



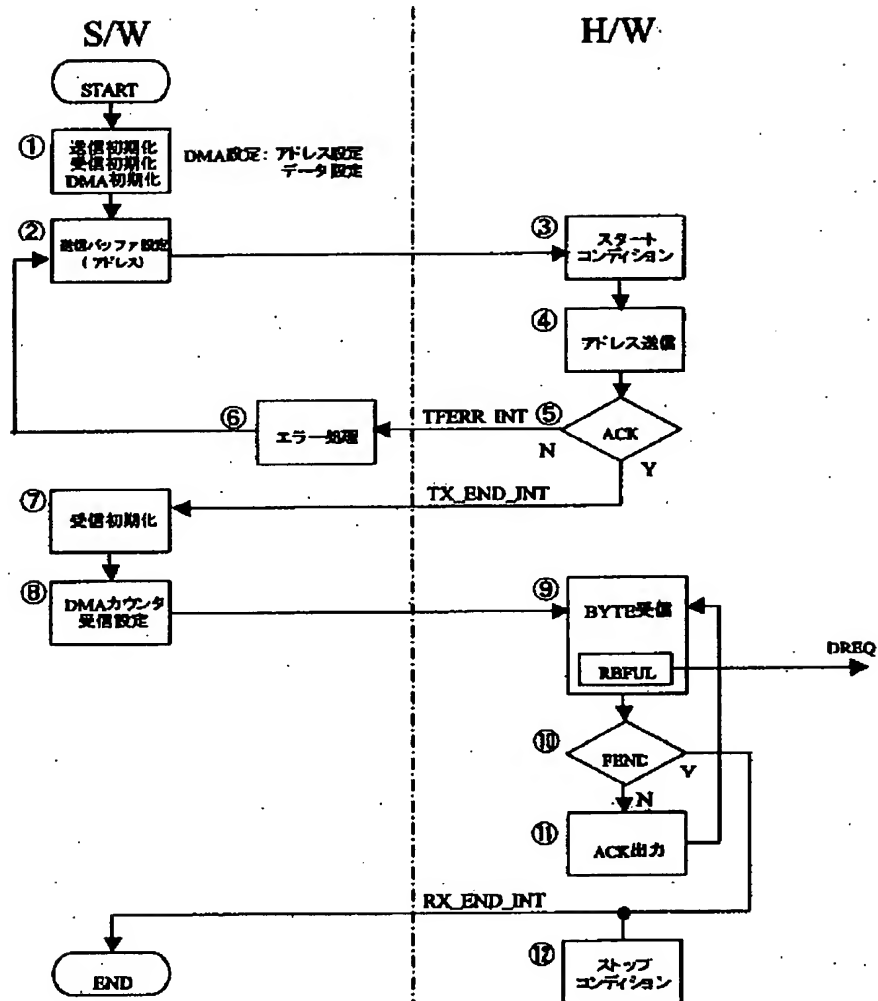
【図39】



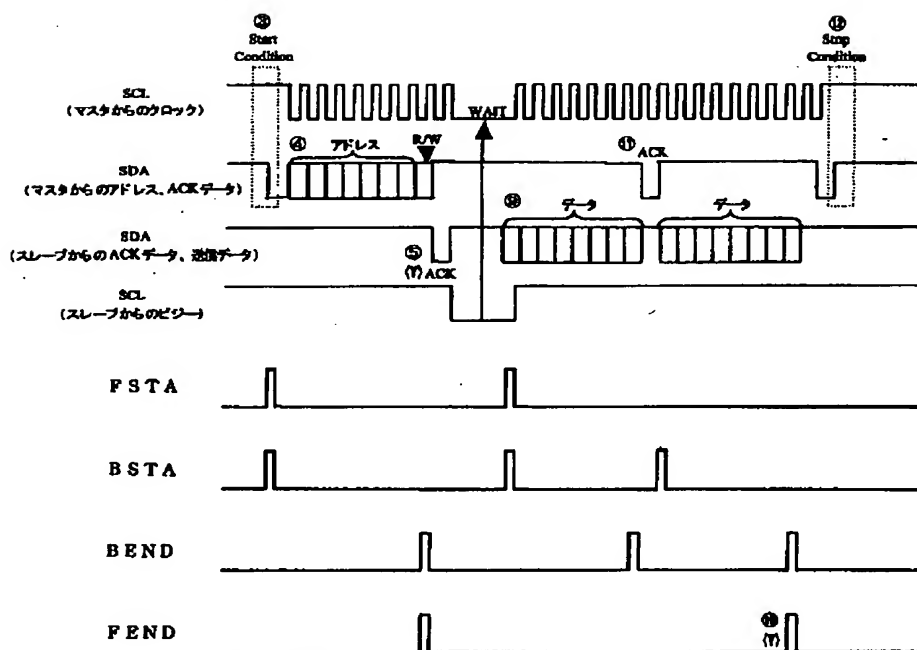
【図47】



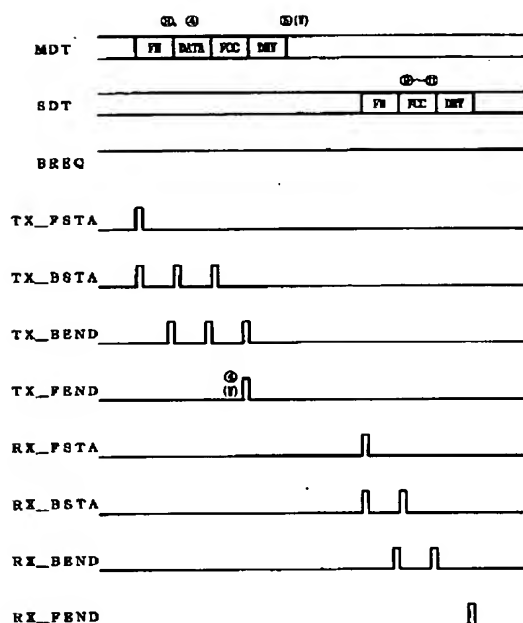
【図37】



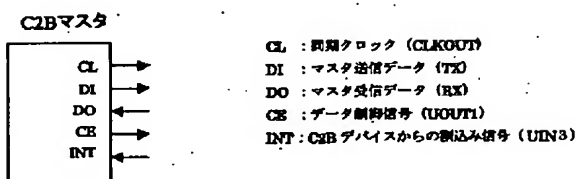
【図38】



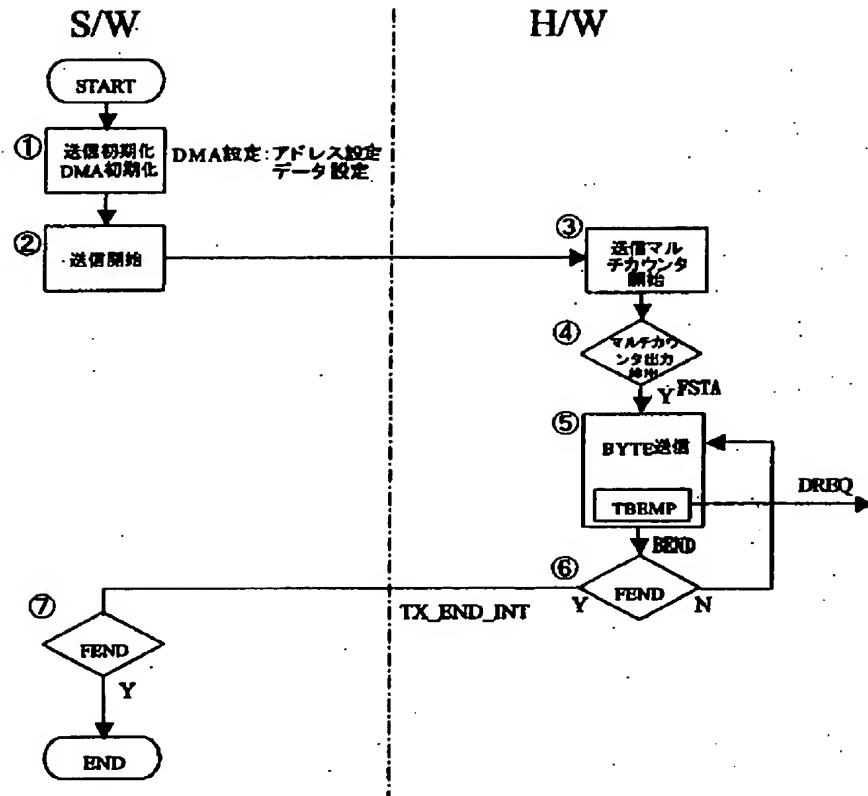
【図50】



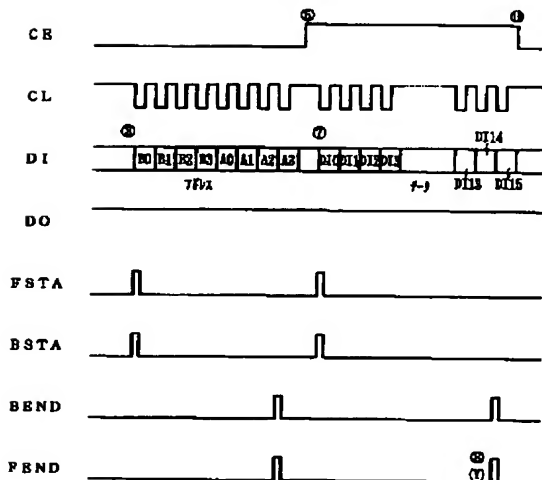
【図53】



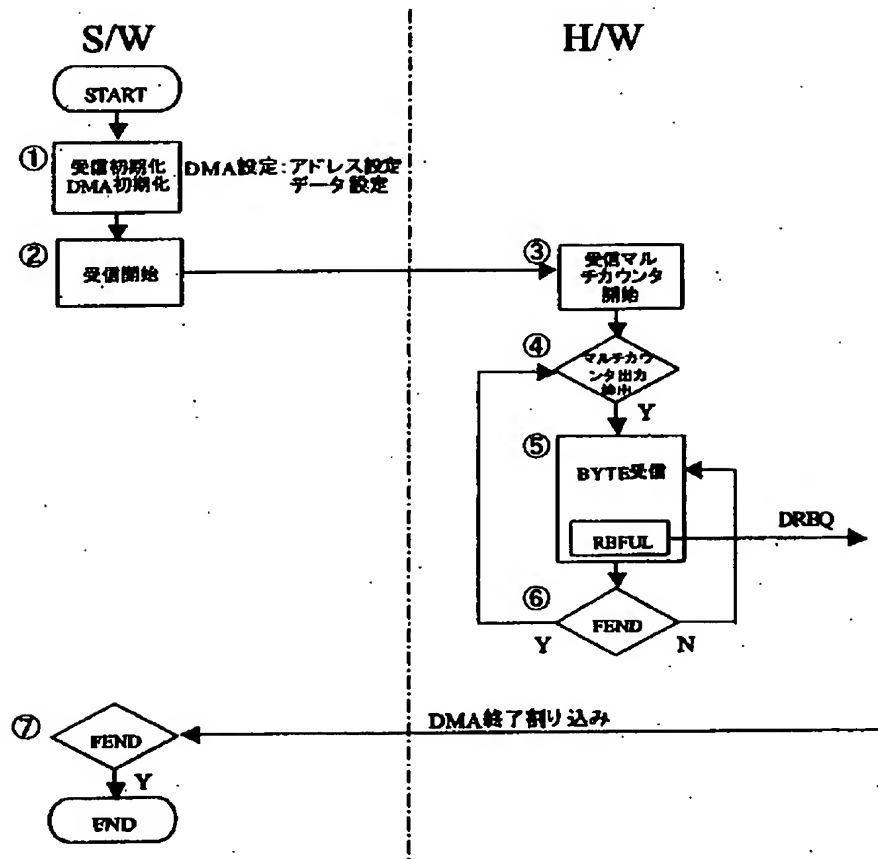
【図40】



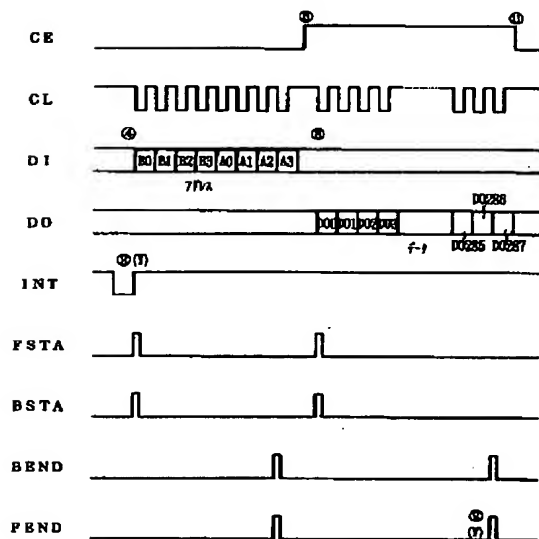
【図55】



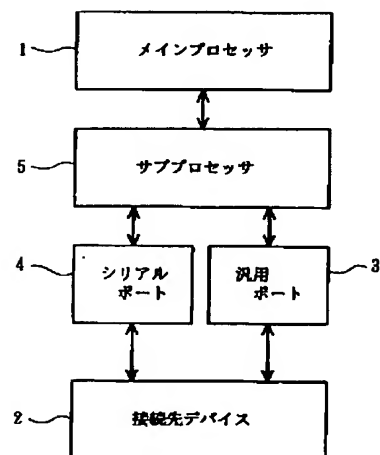
【図42】



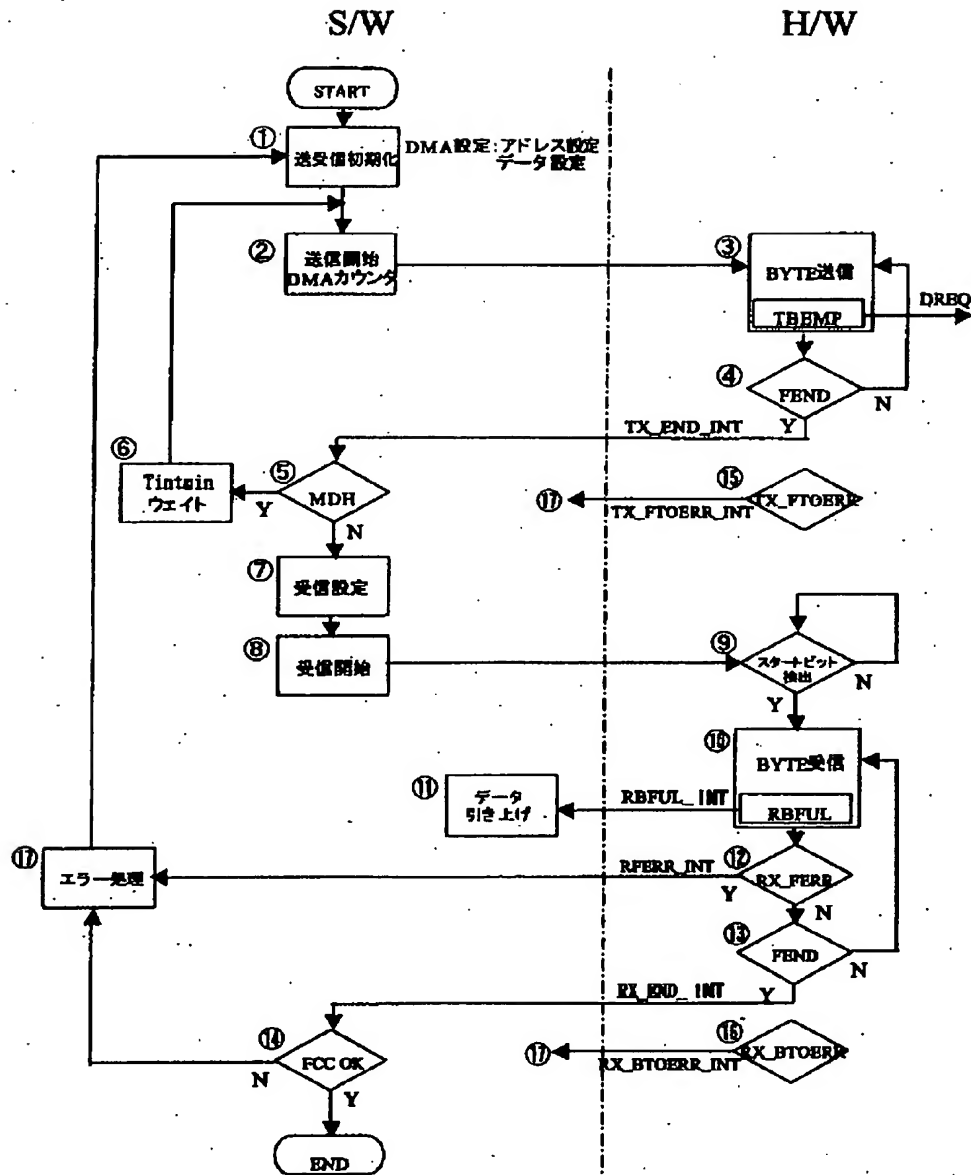
【図57】



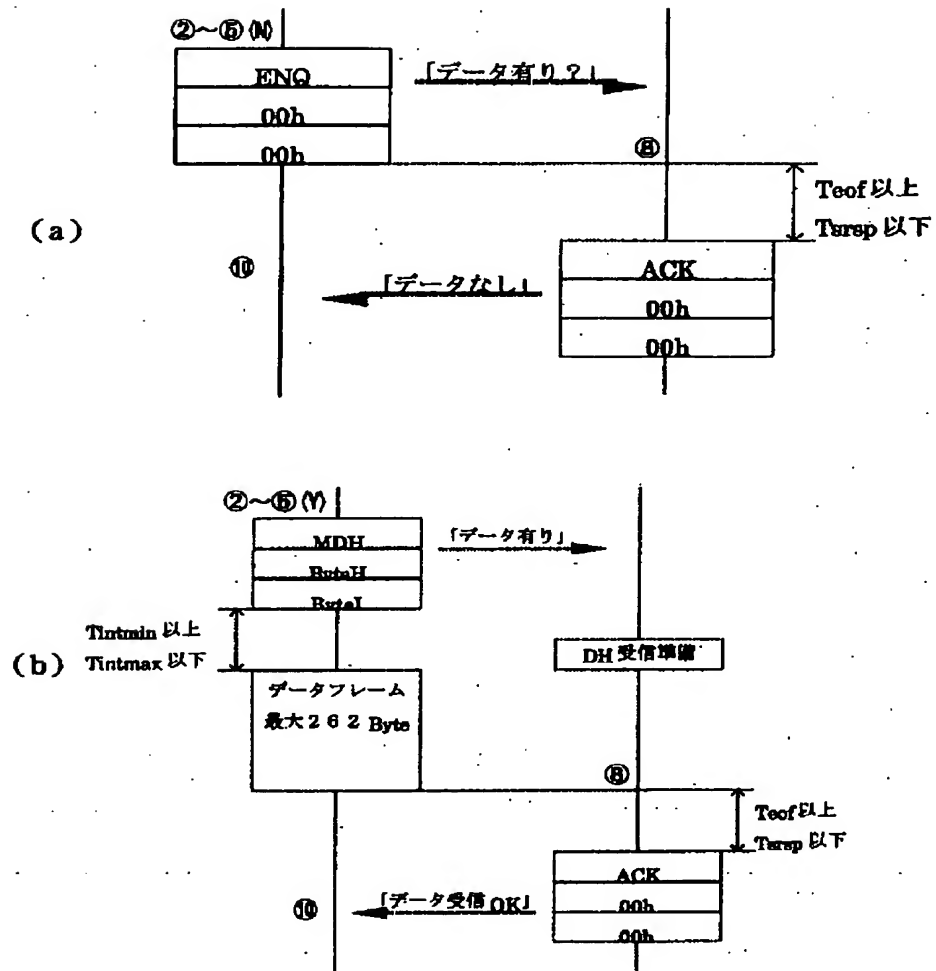
【図58】



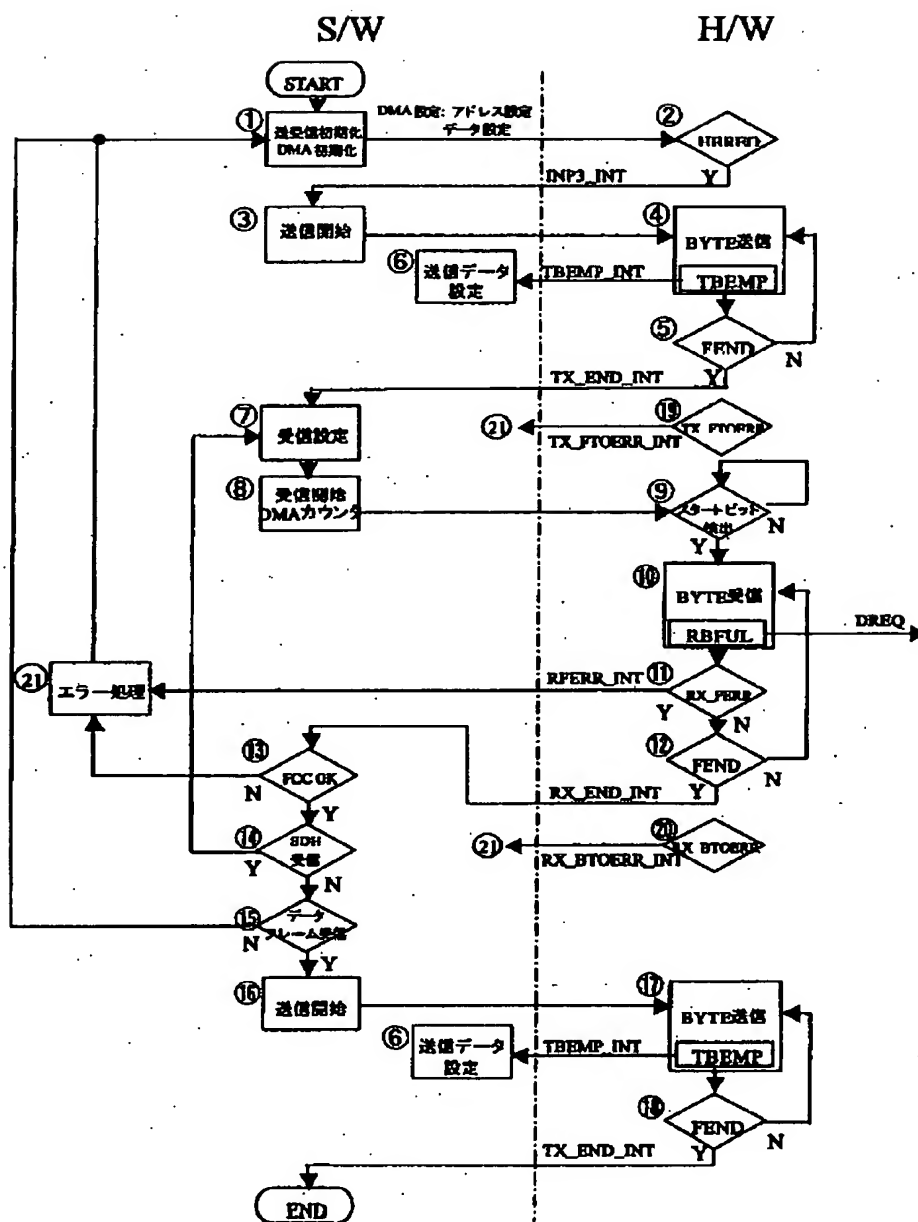
【図44】



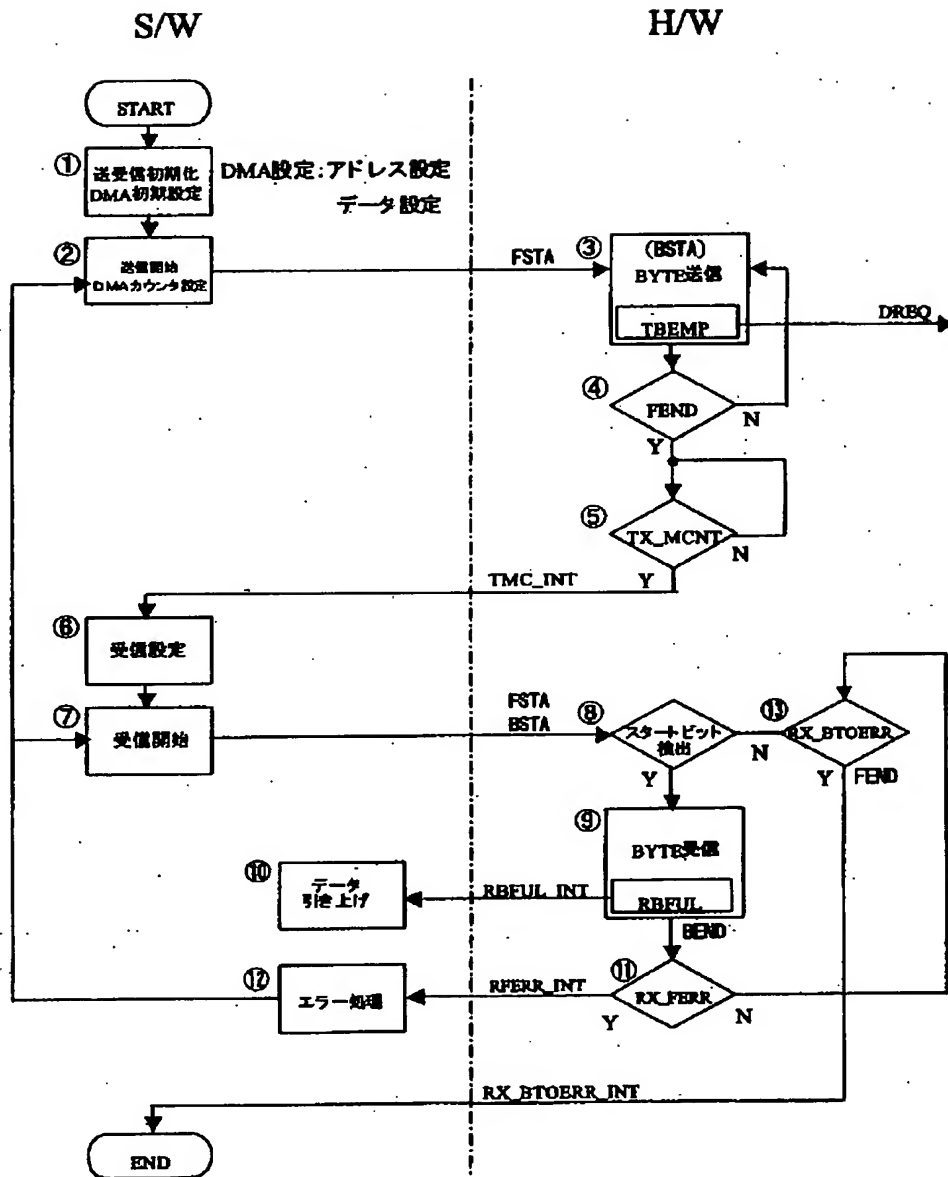
【図45】



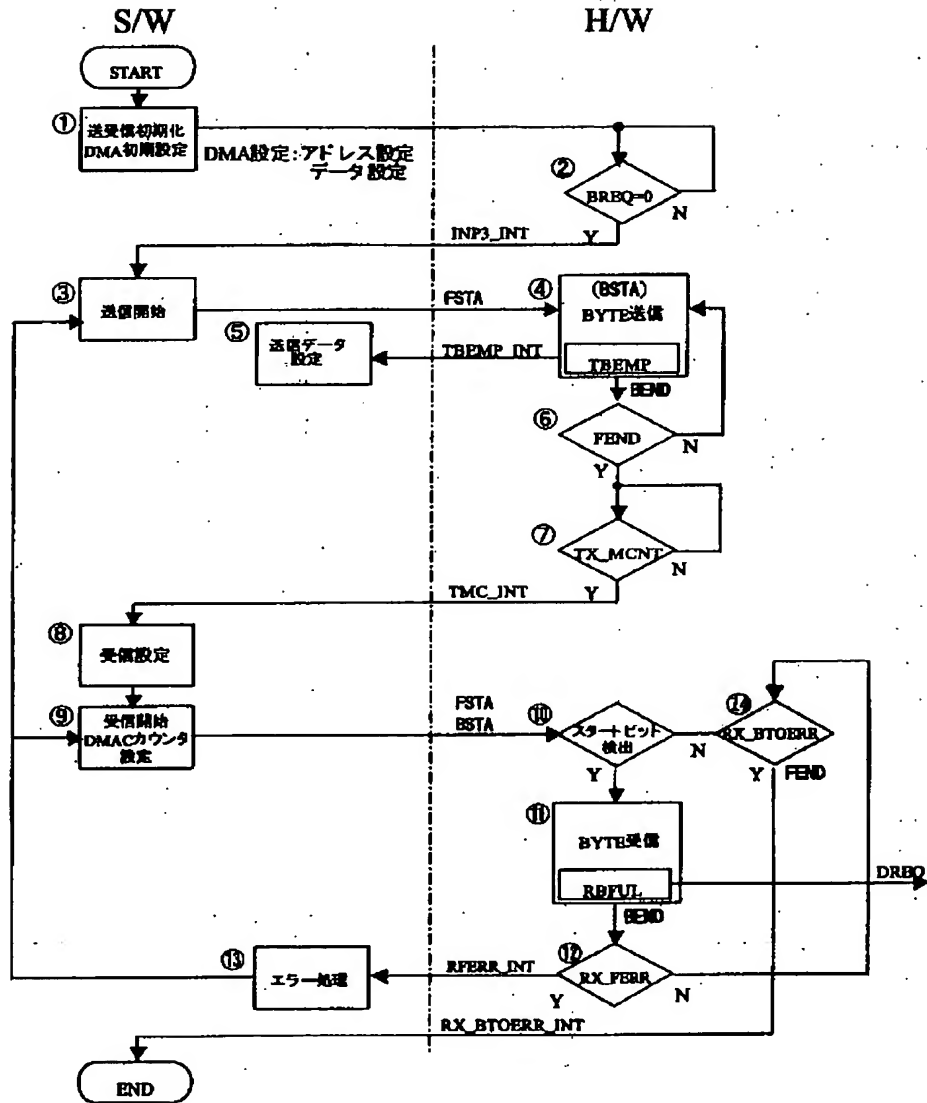
【図 4 6】



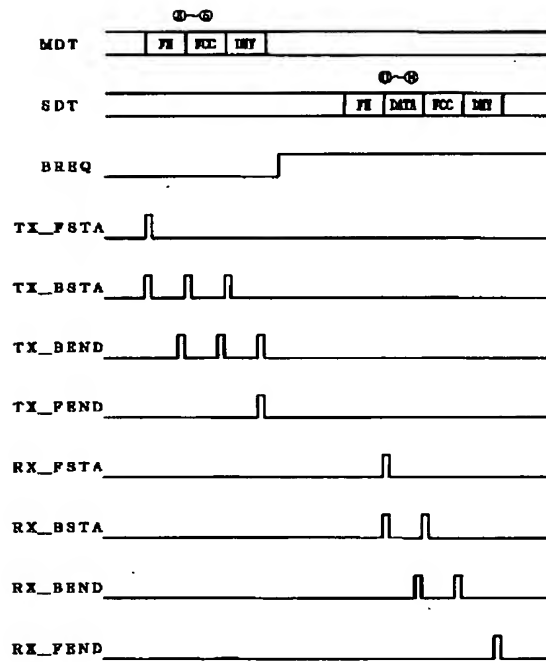
【図49】



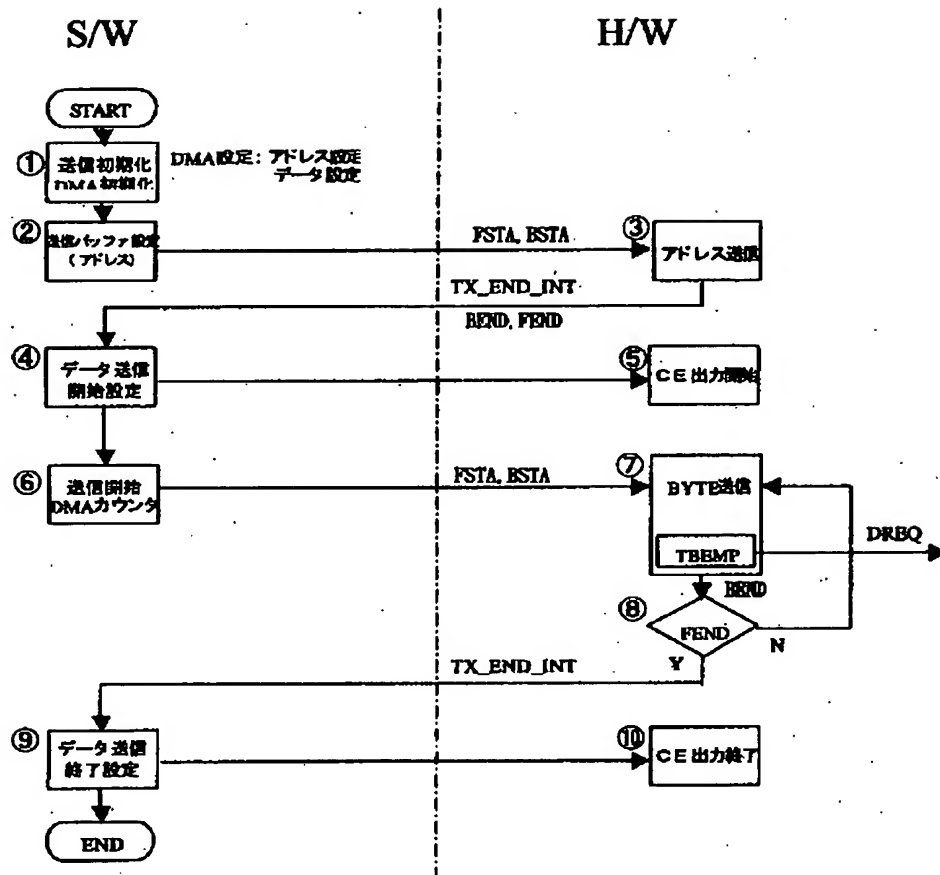
【図51】



【図 52】



【図54】



【図56】

